

Operating System/2™ Programming Tools and Information Version 1.2

Presentation Manager FORTRAN/2 Bindings Reference

64F0281

Operating System/2™
Programming Tools
and Information
Version 1.2

Presentation Manager FORTRAN/2 Bindings Reference

First Edition (September 1989)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

© Copyright International Business Machines Corporation 1986, 1989. All Rights Reserved.

Note to US Government users - Documentation related to Restricted Rights - Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

The following names, used in this publication, are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries:

IBM
Operating System/2
OS/2
Presentation Manager
Systems Application Architecture.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license enquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Armonk, NY 10504.

Library Structure

OS/2 Product

IBM Operating System/2
Standard Edition
Version 1.2

Getting Started
Using Advanced Features
Product Information

6024926 3.5″ diskette
6024930 5.25″ diskette

OS/2 Programming Tools and Information

IBM Operating System/2
Version 1.2

Installation
Programming Overview
Programming Guide
Building Programs
Dialog Tag Language
Guide and Reference
Dialog Manager Guide
and Reference
Dialog Manager and
Dialog Tag Language
Reference Summary
Programming Reference
(3 books)
Bindings Reference for
Presentation Manager
(4 books for COBOL/2,
FORTRAN/2, C/2, and
Macro Assembler/2)
I/O Subsystems and
Device Support
(2 books)
Systems Application
Architecture
Common User Access:
Advanced Interface
Design Guide

6024929

Separate Order (no charge)

Keyboards and Code Pages

6280345

Available Separately

Command Reference

6024928

Service Coordinator's
Guide

15F2214

Programming Languages

IBM Basic Compiler/2
Version 1.0 6280179

IBM Macro Assembler/2
Version 1.0 6280181

IBM Pascal Compiler/2
Version 1.0 6280183

IBM FORTRAN/2
Version 1.02 6280185

IBM COBOL/2
Version 1.0 6280207

IBM C/2
Version 1.1 6280284

Systems Application Architecture

An Overview

GC26-4341

Writing Applications:
A Design Guide

SC26-4362

Common User Access:
Advanced Interface
Design Guide
SC26-4582

Common User Access:
Basic Interface Design
Guide
SC26-4583

Common Programming Interface

C Reference - Level 2

SC09-1308

COBOL Reference

SC26-4354

Dialog Reference

SC26-4356

FORTRAN Reference

SC26-4357

Procedures Language
Reference
SC26-4358

Presentation Reference

SC26-4359

Contents

Chapter 1. Introduction 1-1
FORTRAN Coding Techniques 1-1
Notation Conventions 1-3
Utility Functions 1-4

Chapter 2. Data Types 2-1

Chapter 3. Device Function Calls 3-1

Chapter 4. Graphics Presentation Interface Function Calls 4-1

Chapter 5. Picture Function Calls 5-1

Chapter 6. Profile Function Calls 6-1

Chapter 7. Spooler Function Calls 7-1

Chapter 8. Window Function Calls 8-1

Index X-1

Chapter 1. Introduction

This book describes the Operating System/2 (OS/2) Version 1.2 language binding for IBM FORTRAN/2.

It provides language-dependent information about the functions described in the OS/2 Version 1.2 *Presentation Manager Programming Reference*.

The information in this book, together with the functional descriptions detailed in the *Presentation Manager Programming Reference*, enable the user to generate call statements in FORTRAN.

The following information is provided:

- The syntax of each data type and structure
- The parameter list for each call.

Note: Calls are listed in alphabetic order of C name, as in the *Presentation Manager Programming Reference*.

FORTRAN Coding Techniques

Data Structures

Structures are organized by overlaying their components onto character strings of equivalent length to the structure, using EQUIVALENCE statements provided in INCLUDE files. There is one EQUIVALENCE statement for each structure.

Each INCLUDE file declares one instance of the structure. If the application needs to use two or more instances, it must use a copying technique (see following example), or it must repeat the definition of the structure, changing the names to avoid duplication.

The structure is declared as a character string, and must *always* be referenced as a character string when used as a parameter on a call. The name of each component of a structure is prefixed with the name of the structure datatype. The prefix is abbreviated where necessary so as not to exceed 8 characters.

The INCLUDE file for the *QMSG* structure is shown below:

```
**********************
C
C**** Constant for length of block (bytes)
      INTEGER*4 QMSG_LEN
      PARAMETER (QMSG_LEN=28)
C
C**** Basic block as CHAR string (to copy to/from)
      CHARACTER*(QMSG_LEN) QMSG_BLOCK
C
C**** DECLAREs for structure components
C**** Window handle
      INTEGER*4 QMSG_HWND
C**** Message identity
      INTEGER*4 QMSG_MSGID
C**** Parameter 1
      INTEGER*4 QMSG_PARAM1
C**** Parameter 2
      INTEGER*4 QMSG_PARAM2
C**** Message time
      INTEGER*4 QMSG_TIME
C**** Pointer position when message was generated
      INTEGER*4 QMSG_X
      INTEGER*4 QMSG_Y
C
C**** EQUIVALENCEs for structure components
      EQUIVALENCE(QMSG_HWND  ,QMSG_BLOCK( 1: 4))
      EQUIVALENCE(QMSG_MSGID ,QMSG_BLOCK( 5: 8))
      EQUIVALENCE(QMSG_PARAM1,QMSG_BLOCK( 9:12))
      EQUIVALENCE(QMSG_PARAM2,QMSG_BLOCK(13:16))
      EQUIVALENCE(QMSG_TIME  ,QMSG_BLOCK(17:20))
      EQUIVALENCE(QMSG_X     ,QMSG_BLOCK(21:24))
      EQUIVALENCE(QMSG_Y     ,QMSG_BLOCK(25:28))
```

If only one instance of a structure is required, the instance declared in the INCLUDE can be used directly. If multiple instances are required, the application should declare additional instances as character strings, and then use the copying technique shown in the example below:

```
C**** Declare QMSG structure
      INCLUDE 'QMSG'
C
C**** Declare multiple instances of structure
      CHARACTER*(QMSG_LEN) MY_QMSG1, MY_QMSG2
C
C**** Copy required instance to EQUIVALENCEd instance
      QMSG_BLOCK=MY_QMSG1
C
C**** Modify or use values as required
      QMSG_MSGID=99
C
C**** Copy EQUIVALENCEd instance back to required instance
      MY_QMSG1=QMSG_BLOCK
```

For those structures containing an array with a variable number of elements (for example, the *ACCELTABLE* structure), the INCLUDE file uses a named constant to represent the dimension of the array (for example, ACCELTAB_DIM). The application must specify the dimension of the array by providing INTEGER and PARAMETER statements which define the dimension required, before the INCLUDE of the structure:

```
      INTEGER*4  ACCELTAB_DIM
      PARAMETER (ACCELTAB_DIM=28)
      INCLUDE    'ACCELTAB'
```

Sample statements are contained in the INCLUDE file as comments, identified by "C-->" in columns 1 through 4.

A component of a structure that is itself a structure is declared as a character string of appropriate length, as defined in the INCLUDE file for that structure:

```
CHARACTER*(ACCEL_LEN)     ACCELTAB_ACCEL(ACCELTAB_DIM)
```

The application must include the referenced structure (*ACCEL*) before including the required structure (*ACCELTABLE*).

Miscellaneous definitions

The following definitions are provided in the FORTRAN constants file:

```
INTEGER*4  FALSE
PARAMETER (FALSE=0)

INTEGER*4  TRUE
PARAMETER (TRUE=1)

INTEGER*4  NULL
PARAMETER (NULL=0)
```

Notation Conventions

Interdependent Parameters

Where the size of one parameter is dependent on another, the binding shows the dependent parameter with its size given by a name in lower case; this is the name of the parameter which controls the size. In an actual application, an appropriate literal value must appear here. The dependent parameter must be declared with an explicit size, and the size parameter must not exceed this value.

The following example, (the WinCreateDataStructure call) shows how an array and a character string are represented:

```
C****                          Number of elements
      INTEGER*4 COUNT
C****                          Datatype codes
      INTEGER*4 TYPES(count)
C****                          Length of input structure
      INTEGER*4 INLEN
C****                          Input structure
      CHARACTER*(inlen) INSTRUC
```

In this example, the word 'count' in lower case indicates that the current value of the variable COUNT must not exceed the number of elements declared for the array TYPES. The word 'inlen' in lower case indicates that the current value of the variable INLEN must not exceed the declared length of the character string INSTRUC.

Maximum Name Length

Where necessary, names are truncated to the first 31 characters.

The Data Type Mapping Symbol

For some parameters, the 'comment' text contains a string such as:

```
(STRL→HSTRUCT)
```

This indicates that the standard data type *STRL* maps to the data type *HSTRUCT*, for the parameter indicated.

Utility Functions

Eleven utility functions are provided to assist with the construction and analysis of datatypes.

Structure Utilities

The following utilities allow data structures in the standard format to be created, queried, and destroyed; such structures are always referenced by a handle (*HSTRUCT*):

WinCreateDataStructure

WinModifyDataStructure

WinQueryDataStructure

WinDestroyDataStructure

WinFreeMsg.

Composed-Integer Utilities

The following utilities allow composed 4-byte integers to be set and queried (data types *COMPOSED*, and *FIXED*, and some instances of *MPARAM* and *MRESULT*):

WinSetValue

WinQueryValue.

Bit Utilities

The following utilities allow bit values to be set and queried (data types *BIT8*, *BIT16*, and *BIT32*):

WinSetBits

WinQueryBits

WinSetBitsUnderMask

WinQueryBitsUnderMask.

Chapter 2. Data Types

ACCEL Accelerator structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
ACCEL_BLOCK)
C***     Options
  INTEGER*4 ACCEL_OPTIONS
C***     Key
  INTEGER*4 ACCEL_KEY
C***     Command code
  INTEGER*4 ACCEL_CMD
```

ACCELTABLE Accelerator-table structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
ACCELTAB_BLOCK)
C***     Number of accelerator entries
  INTEGER*4             ACCELTAB_COUNT
C***     Code page for accelerator entries
  INTEGER*4             ACCELTAB_CODEPAGE
C***     Accelerator entries
  CHARACTER*(ACCEL_LEN) ACCELTAB_ACCEL(ACCELTAB_DIM)
```

ARCPARAM Arc-parameters structure.
(Maps via ARCPARAMS)

ARCPARAMS Arc-parameters structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
ARCPARML_BLOCK)
C***     P coefficient
  INTEGER*4 ARCPARML_P
C***     Q coefficient
  INTEGER*4 ARCPARML_Q
C***     R coefficient
  INTEGER*4 ARCPARML_R
C***     S coefficient
  INTEGER*4 ARCPARML_S
```

AREABUNDLE Area-attributes bundle structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
AREABUN_BLOCK)
C***     Area foreground color
  INTEGER*4 AREABUN_COLOR
C***     Area background color
  INTEGER*4 AREABUN_BACKCOLOR
C***     Area foreground-mix mode
  INTEGER*4 AREABUN_MIXMODE
C***     Area background-mix mode
  INTEGER*4 AREABUN_BACKMIXMODE
C***     Pattern set
  INTEGER*4 AREABUN_SET
C***     Pattern symbol
  INTEGER*4 AREABUN_SYMBOL
C***     Pattern reference point
  INTEGER*4 AREABUN_X
  INTEGER*4 AREABUN_Y
```

ATOM

Atom identity.

```
INTEGER*4
```

BANDRECT

Rectangle structure, used for the coordinates of an output band (see DevEscape).

An empty rectangle is one for which **xleft** is greater than **xright**, or **ybottom** is greater than **ytop**.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
BANDRECT_BLOCK)
C***      x coordinate of left edge of rectangle
  INTEGER*4 BANDRECT_XLEFT
C***      y coordinate of bottom edge of rectangle
  INTEGER*4 BANDRECT_YBOTTOM
C***      x coordinate of right edge of rectangle
  INTEGER*4 BANDRECT_XRIGHT
C***      y coordinate of top edge of rectangle
  INTEGER*4 BANDRECT_YTOP
```

BITMAPINFO

Bit-map information structure.

Each bit plane logically contains (**bitmapwidth*bitmapheight*bitcount**) bits, although the actual length can be greater because of padding.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
BITINF_BLOCK)
C***      Length of fixed portion of structure
  INTEGER*4           BITINF_LENGTH
C***      Bit-map width in pels
  INTEGER*4           BITINF_BITMAPWIDTH
C***      Bit-map height in pels
  INTEGER*4           BITINF_BITMAPHEIGHT
C***      Number of bit planes
  INTEGER*4           BITINF_PLANES
C***      Number of bits per pel within a plane
  INTEGER*4           BITINF_BITCOUNT
C***      Array of RGB values
  CHARACTER*(RGB_LEN) BITINF_COLOR(BITINF_DIM)
```

BITMAPINFOHEADER

Bit-map information header structure.

Each bit plane logically contains (**bitmapwidth*bitmapheight*bitcount**) bits, although the actual length can be greater because of padding.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
BITINFH_BLOCK)
C***      Length of structure
  INTEGER*4 BITINFH_LENGTH
C***      Bit-map width in pels
  INTEGER*4 BITINFH_BITMAPWIDTH
C***      Bit-map height in pels
  INTEGER*4 BITINFH_BITMAPHEIGHT
C***      Number of bit planes
  INTEGER*4 BITINFH_PLANES
C***      Number of bits per pel within a plane
  INTEGER*4 BITINFH_BITCOUNT
```

BIT8

Defines eight independent BOOL values.

```
INTEGER*4
```

BIT16

Defines 16 independent BOOL values.

INTEGER*4

BIT32

Defines 32 independent BOOL values.

INTEGER*4

BOOL

Boolean.

The *BOOL* data type is defined such that any nonzero value indicates TRUE, and a zero value indicates FALSE.

The constants TRUE and FALSE have been defined as 1 and 0 respectively. The constant TRUE should never be used in an equality comparison with a *BOOL* as this may result in incorrect results.

When an application needs to pass a TRUE value on a call, it is recommended that it passes the value TRUE, which is 1, and not any nonzero value.

INTEGER*4

BTNCDATA

Button-control-data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
BTNCDATA_BLOCK)
C***     Length of the control data in bytes
  INTEGER*4 BTNCDATA_LENGTH
C***     Check state of button
  INTEGER*4 BTNCDATA_CHECKSTATE
C***     Highlighting state of button
  INTEGER*4 BTNCDATA_HILITE
```

BUFFER

Buffer.
(Maps via STR)

BUNDLE

Bundle data area. It is overlaid by one of these attribute bundle structures:

AREABUNDLE
CHARBUNDLE
IMAGEBUNDLE
LINEBUNDLE
MARKERBUNDLE.

BYTE

Byte.

CHARACTER*1

CHAR

Single-byte character.

A 1-character CHAR maps to CHARACTER*1.

Otherwise an n-character CHAR is represented by the pair of values:

Length: INTEGER*4
String: CHARACTER*(n)

CHARBUNDLE Character-attributes bundle structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
CHARBUN_BLOCK)
C***      Character foreground color
  INTEGER*4 CHARBUN_COLOR
C***      Character background color
  INTEGER*4 CHARBUN_BACKCOLOR
C***      Character foreground-mix mode
  INTEGER*4 CHARBUN_MIXMODE
C***      Character background-mix mode
  INTEGER*4 CHARBUN_BACKMIXMODE
C***      Character set
  INTEGER*4 CHARBUN_SET
C***      Character precision
  INTEGER*4 CHARBUN_PRECISION
C***      Character cell size
  INTEGER*4 CHARBUN_WIDTH
  INTEGER*4 CHARBUN_HEIGHT
C***      Character angle
  INTEGER*4 CHARBUN_X
  INTEGER*4 CHARBUN_Y
C***      Character shear
  INTEGER*4 CHARBUN_X_2
  INTEGER*4 CHARBUN_Y_2
C***      Character direction
  INTEGER*4 CHARBUN_DIRECTION
```

CLASSINFO Class-information structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
CLASSINF_BLOCK)
C***      Class-style flags
  INTEGER*4 CLASSINF_STYLE
C***      Window procedure
  INTEGER*4 CLASSINF_WINDOWPROC
C***      Number of additional window words
  INTEGER*4 CLASSINF_EXTRA
```

COMPOSED A 4-byte value that is composed by the concatenation of values of other data types.

For example, it could be composed of two values, one of which has a data type of *SHORT* and the other a data type of *USHORT*.

```
INTEGER*4
```

COUNT2 Count in the range 0 through 65 535.

```
INTEGER*4
```

COUNT2B Count of bytes, in the range 0 through 65 535.

```
INTEGER*4
```

COUNT2CH Count of characters, in the range 0 through 65 535.

```
INTEGER*4
```

COUNT4 Count in the range 0 through 2 147 483 647.

```
INTEGER*4
```

COUNT4B Count of bytes, in the range 0 through 2 147 483 647.

```
INTEGER*4
```

CPID Code-page identity.

```
INTEGER*4
```

CREATEPARAMS Create parameters structure.

This structure can be used to format the application-defined data area used in the WinCreateDlg, the WinLoadDlg, and the WinDlgBox calls in order to conform to SAA.
(HSTRUCT: Use composition or decomposition functions.)

CREATESTRUCT Create-window data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
CRTSTRUC_BLOCK)
C***      Presentation parameters. (PRESDATA->HSTRUCT)
  INTEGER*4 CRTSTRUC_PRESPARAMS
C***      Control data. (CTLDATA->HSTRUCT)
  INTEGER*4 CRTSTRUC_CTLDATA
C***      Window identifier
  INTEGER*4 CRTSTRUC_WINDOWIDENTITY
C***      Window behind which the window is to be placed
  INTEGER*4 CRTSTRUC_BEHIND
C***      Window owner
  INTEGER*4 CRTSTRUC_OWNER
C***      Window height
  INTEGER*4 CRTSTRUC_CY
C***      Window width
  INTEGER*4 CRTSTRUC_CX
C***      y coordinate of origin
  INTEGER*4 CRTSTRUC_Y
C***      x coordinate of origin
  INTEGER*4 CRTSTRUC_X
C***      Window style
  INTEGER*4 CRTSTRUC_STYLE
C***      Window text. (STRL->HSTRUCT)
  INTEGER*4 CRTSTRUC_TEXT
C***      Registered window class name. (STRL->HSTRUCT)
  INTEGER*4 CRTSTRUC_CLASSNAME
C***      Parent window handle
  INTEGER*4 CRTSTRUC_PARENT
```

CTLDATA Class-specific control data, beginning with a value conforming to a *COUNT2B* data type, which specifies the overall length of the data.

The following are cases of this data type:

BTNCDATA	Button control data.
ENTRYFDATA	Entry field control data.
FRAMECDATA	Frame control data.
SBCDATA	Scroll bar control data.
MLECDATA	Multi-line entry field control data.

(HSTRUCT: Use composition or decomposition functions.)

CURSORINFO Cursor-information structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
CURINF_BLOCK)
C***      Window handle
  INTEGER*4 CURINF_HWND
C***      x coordinate
  INTEGER*4 CURINF_X
C***      y coordinate
  INTEGER*4 CURINF_Y
C***      Cursor width
  INTEGER*4 CURINF_CX
C***      Cursor height
  INTEGER*4 CURINF_CY
C***      Options
  INTEGER*4 CURINF_OPTIONS
C***      Cursor box
  INTEGER*4 CURINF_XLEFT
  INTEGER*4 CURINF_YBOTTOM
  INTEGER*4 CURINF_XRIGHT
  INTEGER*4 CURINF_YTOP
```

DEVOPENDATA Open-device data area. The format of this area is the same as a *DEVOPENSTRUC* structure.
(Maps via DEVOPENSTRUC)

DEVOPENSTRUC Open-device data structure.

The same structure is applicable to both DevOpenDC and SplQmOpen calls.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
DEVSTRUC_BLOCK)
C***      Logical address. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_ADDRESS
C***      Driver name. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_DRIVERNAME
C***      Driver data. (DRIVDATA→HSTRUCT)
  INTEGER*4 DEVSTRUC_DRIVERDATA
C***      Data type. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_DATATYPE
C***      Comment. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_COMMENT
C***      Queue-processor name. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_QUEUEPROCNAME
C***      Queue-processor parameters. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_QUEUEPROCPARAMS
C***      Spooler parameters. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_SPOOLERPARAMS
C***      Network parameters. (STRL→HSTRUCT)
  INTEGER*4 DEVSTRUC_NETWORKPARAMS
```

DLGTEMPLATE Dialog-template structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
DLGT_BLOCK)
C***     Length of template
  INTEGER*4                DLGT_LENGTH
C***     Template format type
  INTEGER*4                DLGT_TYPE
C***     Code page
  INTEGER*4                DLGT_CODEPAGE
C***     Offset to dialog items
  INTEGER*4                DLGT_ITEMOFFSET
C***     Template status
  INTEGER*4                DLGT_TEMPLATESTATUS
C***     Index of item to receive focus initially
  INTEGER*4                DLGT_ITEMFOCUS
C***     Count of presentation-parameter offsets
  INTEGER*4                DLGT_PRESPARAMSOFFSETS
C***     Start of dialog items
  CHARACTER*(DLGTITEM_LEN) DLGT_DLGTI(DLGT_DIM)
```

DLGTITEM Dialog-item structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
DLGTITEM_BLOCK)
C***     Status
  INTEGER*4 DLGTITEM_ITEMSTATUS
C***     Count of children to this dialog item
  INTEGER*4 DLGTITEM_CHILDREN
C***     Length of class name
  INTEGER*4 DLGTITEM_CLASSLEN
C***     Offset to class name
  INTEGER*4 DLGTITEM_CLASSNAME
C***     Length of text
  INTEGER*4 DLGTITEM_TEXTLEN
C***     Offset to text
  INTEGER*4 DLGTITEM_TEXT
C***     Dialog item window style
  INTEGER*4 DLGTITEM_STYLE
C***     x coordinate of origin of dialog-item window
  INTEGER*4 DLGTITEM_X
C***     y coordinate of origin of dialog-item window
  INTEGER*4 DLGTITEM_Y
C***     Dialog-item window width
  INTEGER*4 DLGTITEM_WIDTH
C***     Dialog-item window height
  INTEGER*4 DLGTITEM_HEIGHT
C***     Identity
  INTEGER*4 DLGTITEM_ID
C***     Reserved
  INTEGER*4 DLGTITEM_PRESPARAMS
C***     Offset to control data
  INTEGER*4 DLGTITEM_CTLDATA
```

DRIVDATA Driver-data structure.
(HSTRUCT: Use composition or decomposition functions.)

ENTRYFDATA Entry field control data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
ENTFDATA_BLOCK)
C***      Length of control data in bytes
  INTEGER*4 ENTFDATA_LENGTH
C***      Edit limit
  INTEGER*4 ENTFDATA_EDITLIMIT
C***      Minimum selection
  INTEGER*4 ENTFDATA_MINSEL
C***      Maximum selection
  INTEGER*4 ENTFDATA_MAXSEL
C***      Picture mask string. (STRL→HSTRUCT)
```

ERRINFO Error-information structure.
(HSTRUCT: Use composition or decomposition functions.)

ERRORID Error identity.

```
INTEGER*4
```

FATTRS Font-attributes structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
FATTRS_BLOCK)
C***      Length of record
  INTEGER*4    FATTRS_LENGTH
C***      Selection indicators
  INTEGER*4    FATTRS_SELECTION
C***      Matched-font identity
  INTEGER*4    FATTRS_MATCH
C***      Typeface name
  INTEGER*4    FATTRS_LENGTH_2
  CHARACTER*32 FATTRS_FACENAME
C***      Registry identifier
  INTEGER*4    FATTRS_REGISTRY
C***      Code page
  INTEGER*4    FATTRS_CODEPAGE
C***      Maximum base-line extension
  INTEGER*4    FATTRS_MAXBASELINEEXT
C***      Average character width
  INTEGER*4    FATTRS_AVECHARWIDTH
C***      Type indicators
  INTEGER*4    FATTRS_TYPE
C***      Font-use indicators
  INTEGER*4    FATTRS_FONTUSE
```

FFDESCS Font-file descriptor.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
FFDESCS_BLOCK)
C***      Family name
  INTEGER*4    FFDESCS_LENGTH
  CHARACTER*32 FFDESCS_FAMILYNAME
C***      Face name
  INTEGER*4    FFDESCS_LENGTH_2
  CHARACTER*32 FFDESCS_FACENAME
```

FIXED

Signed-integer fraction (16:16). This is a concatenation of a SHORT with a USHORT.
(Maps via COMPOSED)

FONTMETRICS

Font-metrics structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
FMETRS_BLOCK)
 C***      Family name
   INTEGER*4     FMETRS_LENGTH
   CHARACTER*32 FMETRS_FAMILYNAME
 C***      Facename
   INTEGER*4     FMETRS_LENGTH_2
   CHARACTER*32 FMETRS_FACENAME
 C***      Registry identifier
   INTEGER*4     FMETRS_REGISTRY
 C***      Code page
   INTEGER*4     FMETRS_CODEPAGE
 C***      Em height
   INTEGER*4     FMETRS_EMHEIGHT
 C***      x height
   INTEGER*4     FMETRS_XHEIGHT
 C***      Maximum ascender
   INTEGER*4     FMETRS_MAXASCENDER
 C***      Maximum descender
   INTEGER*4     FMETRS_MAXDESCENDER
 C***      Lowercase ascent
   INTEGER*4     FMETRS_LOWERCASEASCENT
 C***      Lowercase descent
   INTEGER*4     FMETRS_LOWERCASEDESCENT
 C***      Internal leading
   INTEGER*4     FMETRS_INTERNALLEADING
 C***      External leading
   INTEGER*4     FMETRS_EXTERNALLEADING
 C***      Average character width
   INTEGER*4     FMETRS_AVECHARWIDTH
 C***      Maximum character increment
   INTEGER*4     FMETRS_MAXCHARINC
 C***      Em increment
   INTEGER*4     FMETRS_EMINC
 C***      Maximum baseline extent
   INTEGER*4     FMETRS_MAXBASELINEEXT
 C***      Character slope
   INTEGER*4     FMETRS_CHARSLOPE
 C***      Inline direction
   INTEGER*4     FMETRS_INLINEDIR
 C***      Character rotation
   INTEGER*4     FMETRS_CHARROT
 C***      Weight class
   INTEGER*4     FMETRS_WEIGHTCLASS
 C***      Width class
   INTEGER*4     FMETRS_WIDTHCLASS
 C***      x device resolution
   INTEGER*4     FMETRS_XDEVICERES
 C***      y device resolution
   INTEGER*4     FMETRS_YDEVICERES
 C***      First character
   INTEGER*4     FMETRS_FIRSTCHAR
```

```
C***      Last character
  INTEGER*4    FMETRS_LASTCHAR
C***      Default character
  INTEGER*4    FMETRS_DEFAULTCHAR
C***      Break character
  INTEGER*4    FMETRS_BREAKCHAR
C***      Nominal point size
  INTEGER*4    FMETRS_NOMINALPOINTSIZE
C***      Minimum point size
  INTEGER*4    FMETRS_MINIMUMPOINTSIZE
C***      Maximum point size
  INTEGER*4    FMETRS_MAXIMUMPOINTSIZE
C***      Type indicators
  INTEGER*4    FMETRS_TYPE
C***      Definition indicators
  INTEGER*4    FMETRS_DEFN
C***      Selection indicators
  INTEGER*4    FMETRS_SELECTION
C***      Capabilities
  INTEGER*4    FMETRS_CAPABILITIES
C***      Subscript x size
  INTEGER*4    FMETRS_SUBSCRIPTXSIZE
C***      Subscript y size
  INTEGER*4    FMETRS_SUBSCRIPTYSIZE
C***      Subscript x offset
  INTEGER*4    FMETRS_SUBSCRIPTXOFFSET
C***      Subscript y offset
  INTEGER*4    FMETRS_SUBSCRIPTYOFFSET
C***      Superscript x size
  INTEGER*4    FMETRS_SUPERSCRIPTXSIZE
C***      Superscript y size
  INTEGER*4    FMETRS_SUPERSCRIPTYSIZE
C***      Superscript x offset
  INTEGER*4    FMETRS_SUPERSCRIPTXOFFSET
C***      Superscript y offset
  INTEGER*4    FMETRS_SUPERSCRIPTYOFFSET
C***      Underscore size
  INTEGER*4    FMETRS_UNDERSCORESIZE
C***      Underscore position
  INTEGER*4    FMETRS_UNDERSCOREPOSITION
C***      Strikeout size
  INTEGER*4    FMETRS_STRIKEOUTSIZE
C***      Strikeout position
  INTEGER*4    FMETRS_STRIKEOUTPOSITION
C***      Kerning pairs
  INTEGER*4    FMETRS_KERNINGPAIRS
C***      Family class
  INTEGER*4    FMETRS_FAMILYCLASS
C***      Matched font identity
  INTEGER*4    FMETRS_MATCH
```

FRAMECDATA Frame-control-data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
FRACDATA_BLOCK)
C***      Length
  INTEGER*4 FRACDATA_LENGTH
C***      Frame-creation flags
  INTEGER*4 FRACDATA_CREATEFLAGS
C***      Identifier of required resource
  INTEGER*4 FRACDATA_RESOURCES
C***      Resource identifier
  INTEGER*4 FRACDATA_RESOURCE
```

GRADIENT Direction-vector structure.
(Maps via GRADIENTL)

GRADIENTL Direction-vector structure.
It is represented by the pair of values:
X, Y: INTEGER*4

HAB Anchor-block handle.
(Maps via HANDLE)

HACCEL Accelerator-table handle.
(Maps via HANDLE)

HANDLE The handle of a resource.
INTEGER*4

HAPP Handle of an application started by the WinInstStartApp function.
(Maps via HANDLE)

HATOMTBL Atom-table handle.
(Maps via HANDLE)

HBITMAP Bit-map handle.
(Maps via HANDLE)

HCINFO Hardcopy-capabilities structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
HCINFO_BLOCK)
C***      Form name
  INTEGER*4    HCINFO_LENGTH
  CHARACTER*32 HCINFO_FORMNAME
C***      Width (left-to-right) in millimeters
  INTEGER*4    HCINFO_XWIDTH
C***      Height (top-to-bottom) in millimeters
  INTEGER*4    HCINFO_YHEIGHT
C***      Left clip limit in millimeters
  INTEGER*4    HCINFO_XLEFTCLIP
C***      Bottom clip limit in millimeters
  INTEGER*4    HCINFO_YBOTTOMCLIP
C***      Right clip limit in millimeters
  INTEGER*4    HCINFO_XRIGHTCLIP
C***      Top clip limit in millimeters
  INTEGER*4    HCINFO_YTOPCLIP
C***      Number of pels between left and right clip limits
  INTEGER*4    HCINFO_XPELS
C***      Number of pels between bottom and top clip limits
  INTEGER*4    HCINFO_YPELS
C***      Attributes of the form identifier
  INTEGER*4    HCINFO_ATTRIBUTES
```

HDC Device-context handle.
(Maps via HANDLE)

HENUM Window-enumeration handle.
(Maps via HANDLE)

HINI Initialization-file handle.
(Maps via HANDLE)

HLIB Library handle.
(Maps via HANDLE)

HMF Metafile handle.
(Maps via HANDLE)

HMODULE Module handle.
```
INTEGER*4
```

HMQ Message-queue handle.
(Maps via HANDLE)

HPOINTER Pointer handle.
(Maps via HANDLE)

HPROC Processor handle.
(Maps via HANDLE)

HPROGARRAY Array of program handles.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
HPROGA_BLOCK)
C***      Program handle array
  INTEGER*4 HPROGA_AHPROG(HPROGA_DIM)
```

HPROGRAM Program handle.
(Maps via HANDLE)

HPS Presentation-space handle.
(Maps via HANDLE)

HRGN Region handle.
(Maps via HANDLE)

HSEM Semaphore handle.
```
INTEGER*4
```

HSPL Spooler handle.
(Maps via HANDLE)

HSTRUCT Structure handle.

An *HSTRUCT* must *always* be given a value by the application prior to its use on a call or in a structure. This applies even when the *HSTRUCT* or structure is an output parameter.

The *HSTRUCT* should be set to the NULL value for those calls and structures where it is not required to have been set to the handle of a previously-created data structure.
```
INTEGER*4
```

HSWITCH Switch-list-entry handle.
(Maps via HANDLE)

HVPS VIO presentation-space handle.
```
INTEGER*4
```

HWND Window handle.
(Maps via HANDLE)

IDENTITY Identity value. Up to 65 536 different identities are available.

```
INTEGER*4
```

IDENTITY4 Identity value. Up to 4 294 967 296 different identities are available.

```
INTEGER*4
```

IMAGEBUNDLE Image-attributes bundle structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
IMAGEBUN_BLOCK)
C***      Image foreground color
  INTEGER*4 IMAGEBUN_COLOR
C***      Image background color
  INTEGER*4 IMAGEBUN_BACKCOLOR
C***      Image foreground-mix mode
  INTEGER*4 IMAGEBUN_MIXMODE
C***      Image background-mix mode
  INTEGER*4 IMAGEBUN_BACKMIXMODE
```

INDEX2 Index value, in the range 0 through 65 535.

```
INTEGER*4
```

IPT Insertion point for multi-line entry field.
(Maps via LONG)

KERNINGPAIRS Kerning-pair records structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
KERNPAIR_BLOCK)
C***      First character of pair
  INTEGER*4 KERNPAIR_FIRSTCHAR
C***      Second character of pair
  INTEGER*4 KERNPAIR_SECONDCHAR
C***      Amount of kerning for this pair
  INTEGER*4 KERNPAIR_KERNINGAMOUNT
```

LENGTH1 Length value, in the range 0 through 255.

```
INTEGER*4
```

LENGTH2 Length value, in the range 0 through 65 535.

```
INTEGER*4
```

LENGTH4 Length value, in the range 0 through 2 147 483 647.

```
INTEGER*4
```

LINEBUNDLE Line-attributes bundle structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
LINEBUN_BLOCK)
C***      Line foreground color
  INTEGER*4 LINEBUN_COLOR
C***      Reserved
  INTEGER*4 LINEBUN_BACKCOLOR
C***      Line foreground-mix mode
  INTEGER*4 LINEBUN_MIXMODE
C***      Reserved
  INTEGER*4 LINEBUN_BACKMIXMODE
C***      Line width
  INTEGER*4 LINEBUN_WIDTH
C***      Geometric line width
  INTEGER*4 LINEBUN_GEOMWIDTH
C***      Line type
  INTEGER*4 LINEBUN_TYPE
C***      Line end
  INTEGER*4 LINEBUN_END
C***      Line join
  INTEGER*4 LINEBUN_JOIN
```

LONG Signed integer in the range –2 147 483 648 through 2 147 483 647.

Note: Where this data type represents a graphic coordinate in world or model space, its value is restricted to –134 217 728 through 134 217 727. A graphic coordinate in device or screen coordinates is restricted to –32 768 through 32 767. Its value may be further restricted by any transforms currently in force, including the positioning of the origin of the window on the screen. In particular, coordinates in world or model space must not generate coordinate values after transformation (that is in device or screen space) outside the range –32 768 through 32 767.

```
INTEGER*4
```

MARGSTRUCT Multi-line entry field margin information

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
MARGST_BLOCK)
   INTEGER*4 MARGST_FLAGS
   INTEGER*4 MARGST_MOUMSG
   INTEGER*4 MARGST_NEAR
```

MARKERBUNDLE Marker-attributes bundle structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
MARKBUN_BLOCK)
 C***      Marker foreground color
   INTEGER*4 MARKBUN_COLOR
 C***      Marker background color
   INTEGER*4 MARKBUN_BACKCOLOR
 C***      Marker foreground-mix mode
   INTEGER*4 MARKBUN_MIXMODE
 C***      Marker background-mix mode
   INTEGER*4 MARKBUN_BACKMIXMODE
 C***      Marker set
   INTEGER*4 MARKBUN_SET
 C***      Marker symbol
   INTEGER*4 MARKBUN_SYMBOL
 C***      Marker cell
   INTEGER*4 MARKBUN_WIDTH
   INTEGER*4 MARKBUN_HEIGHT
```

MATRIX Matrix-elements structure.
(Maps via MATRIXLF)

MATRIXLF Matrix-elements structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
MATRIXLF_BLOCK)
 C***      First element of first row
   INTEGER*4 MATRIXLF_M11
 C***      Second element of first row
   INTEGER*4 MATRIXLF_M12
 C***      Third element of first row
   INTEGER*4 MATRIXLF_M13
 C***      First element of second row
   INTEGER*4 MATRIXLF_M21
 C***      Second element of second row
   INTEGER*4 MATRIXLF_M22
 C***      Third element of second row
   INTEGER*4 MATRIXLF_M23
 C***      First element of third row
   INTEGER*4 MATRIXLF_M31
 C***      Second element of third row
   INTEGER*4 MATRIXLF_M32
 C***      Third element of third row
   INTEGER*4 MATRIXLF_M33
```

MENUITEM Menu item.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
MENUITEM_BLOCK)
C***      Position
  INTEGER*4 MENUITEM_POSITION
C***      Style
  INTEGER*4 MENUITEM_STYLE
C***      Attribute
  INTEGER*4 MENUITEM_ATTRIBUTE
C***      Identity
  INTEGER*4 MENUITEM_IDENTITY
C***      Submenu
  INTEGER*4 MENUITEM_SUBMENU
C***      Item
  INTEGER*4 MENUITEM_ITEM
```

MLECTLDATA Multi-line entry-field control data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
MLECDATA_BLOCK)
C***      Length of control data in bytes
  INTEGER*4 MLECDATA_LENGTH
C***      Import/export format
  INTEGER*4 MLECDATA_FORMAT
C***      Text limit
  INTEGER*4 MLECDATA_TEXTLIMIT
C***      Selection anchor point
  INTEGER*4 MLECDATA_ANCHOR
C***      Selection cursor point
  INTEGER*4 MLECDATA_CURSOR
C***      Formatting rectangle width in pixels
  INTEGER*4 MLECDATA_XFORMAT
C***      Formatting rectangle height in pixels
  INTEGER*4 MLECDATA_YFORMAT
C***      Format flags
  INTEGER*4 MLECDATA_FLAGS
```

MLE_SEARCHDATA Search structure for multi-line entry field.
(HSTRUCT: Use composition or decomposition functions.)

MPARAM

4-byte message-dependent parameter structure.

Certain elements of information, placed into the parameters of a message, have data types that do not consume the whole of the 4 bytes of this data type. The rules governing these cases are:

BOOL The value is contained in the low word and the high word is zero.

SHORT The value is contained in the low word and its sign is extended into the high word.

USHORT The value is contained in the low word and the high word is zero.

NULL The entire 4 bytes are zero.

This datatype can be a *HSTRUCT* or a *COMPOSED* data type.

```
INTEGER*4
```

MRESULT

4-byte message-dependent reply parameter structure.

Certain elements of information, placed into the parameters of a message, have data types that do not consume the whole of the 4 bytes of this data type. The rules governing these cases are:

BOOL The value is contained in the low word and the high word is zero.

SHORT The value is contained in the low word and its sign is extended into the high word.

USHORT The value is contained in the low word and the high word is zero.

NULL The entire 4 bytes are zero.

This datatype can be a *HSTRUCT* or a *COMPOSED* datatype.

```
INTEGER*4
```

MQINFO

Message-queue information structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
MQINF_BLOCK)
 C***      Length of structure
   INTEGER*4 MQINF_B
 C***      Process identity
   INTEGER*4 MQINF_PID
 C***      Thread identity
   INTEGER*4 MQINF_TID
 C***      Message count
   INTEGER*4 MQINF_MSGS
 C***      Reserved. (STORAGE->HSTRUCT)
   INTEGER*4 MQINF_RESERVED
```

OFFSET2B

Offset value in bytes, in the range 0 through 65 535.

```
INTEGER*4
```

OVERFLOW

Overflow error structure for multi-line entry field.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
OVERFLOW_BLOCK)
 C***      One or more EFR_* flags
   INTEGER*4 OVERFLOW_ERRIND
 C***      Number of bytes over the limit
   INTEGER*4 OVERFLOW_BYTESOVER
 C***      Number of pixels over the horizontal limit
   INTEGER*4 OVERFLOW_HORZOVER
 C***      Number of pixels over the vertical limit
   INTEGER*4 OVERFLOW_VERTOVER
```

OWNERITEM Owner item.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
OWNERIT_BLOCK)
C***      Window handle
  INTEGER*4 OWNERIT_HWND
C***      Presentation-space handle
  INTEGER*4 OWNERIT_HPS
C***      State
  INTEGER*4 OWNERIT_STATE
C***      Attribute
  INTEGER*4 OWNERIT_ATTRIBUTE
C***      Old state
  INTEGER*4 OWNERIT_STATEOLD
C***      Old attribute
  INTEGER*4 OWNERIT_ATTRIBUTEOLD
C***      Item rectangle
  INTEGER*4 OWNERIT_XLEFT
  INTEGER*4 OWNERIT_YBOTTOM
  INTEGER*4 OWNERIT_XRIGHT
  INTEGER*4 OWNERIT_YTOP
C***      Item identity
  INTEGER*4 OWNERIT_IDENTITY
C***      Item
  INTEGER*4 OWNERIT_ITEM
```

PARAM Presentation parameter attribute definition.
(HSTRUCT: Use composition or decomposition functions.)

PIBSTRUCT Program-information-block structure.
(HSTRUCT: Use composition or decomposition functions.)

PID Process identity.
INTEGER*4

PIX Pel count for multi-line entry field.
(Maps via LONG)

POINT Point structure.
(Maps via POINTL)

POINTL Point structure (long integer).
It is represented by the pair of values:
X, Y: INTEGER*4

POINTS Point structure (short integer).
(Maps via COMPOSED)

PRESPARAMS Presentation parameter data.
(HSTRUCT: Use composition or decomposition functions.)

PRESDATA Class-specific control presentation data, conforming to a *PRESPARAM* data type.
(HSTRUCT: Use composition or decomposition functions.)

PRFPROFILE Profile structure.
(HSTRUCT: Use composition or decomposition functions.)

PROC Procedure identifier.
INTEGER*4

PROGCATEGORY Program category.
INTEGER*4

PROGDETAILS Program details structure.
(HSTRUCT: Use composition or decomposition functions.)

PROGRAMENTRY Program-entry structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
PROGENTR_BLOCK)
C***      Program handle
  INTEGER*4                 PROGENTR_HPROG
C***      Program type
  CHARACTER*(PROGTYPE_LEN) PROGENTR_PROGRAMTYPE
C***      Program title (null terminated)
  INTEGER*4                 PROGENTR_LENGTH
  CHARACTER*61              PROGENTR_PROGRAMTITLE
```

PROGTITLE Program-title structure.
(HSTRUCT: Use composition or decomposition functions.)

PROGTYPE Program-type structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
PROGTYPE_BLOCK)
C***      Program category
  INTEGER*4 PROGTYPE_PROGC
C***      Visibility attribute
  INTEGER*4 PROGTYPE_VISIBLE
```

PROPERTY2 Property value. Up to 65 536 different properties are available.

```
INTEGER*4
```

PROPERTY4 Property value. Up to 2 147 483 647 different properties are available.

```
INTEGER*4
```

POINTERINFO Pointer-information structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
PTRINF_BLOCK)
C***      Bit-map size indicator
  INTEGER*4 PTRINF_FLAG
C***      x coordinate of action point
  INTEGER*4 PTRINF_XHOTSPOT
C***      y coordinate of action point
  INTEGER*4 PTRINF_YHOTSPOT
C***      Bit-map handle of pointer
  INTEGER*4 PTRINF_HBITMAP
C***      Bit-map handle of color bit map
  INTEGER*4 PTRINF_COLOR
```

QMOPENDATA Open queue-manager data area. The format of this area is the same as a *DEVOPENSTRUC* structure.
(Maps via DEVOPENSTRUC)

QMSG Message structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
QMSG_BLOCK)
C***      Window handle
  INTEGER*4 QMSG_HWND
C***      Message identity
  INTEGER*4 QMSG_MSGID
C***      Parameter 1
  INTEGER*4 QMSG_PARAM1
C***      Parameter 2
  INTEGER*4 QMSG_PARAM2
C***      Message time
  INTEGER*4 QMSG_TIME
C***      Pointer position when message was generated
  INTEGER*4 QMSG_X
  INTEGER*4 QMSG_Y
```

RECT Rectangle structure.

Unless otherwise stated, points on the right-hand or top boundaries of the rectangle are not considered to be included in the rectangle; points on the left-hand or bottom boundaries (that are not also on the right-hand or top boundaries) are included in the rectangle.
(Maps via RECTL)

RECTL Rectangle structure.
It is represented by the 4 values:

Xleft:	INTEGER*4
Ybottom:	INTEGER*4
Xright:	INTEGER*4
Ytop:	INTEGER*4

RESID Resource identity.

```
INTEGER*4
```

RGB RGB color value.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
RGB_BLOCK)
C***      Blue component of the color definition
  CHARACTER*1 RGB_BLUE
C***      Green component of the color definition
  CHARACTER*1 RGB_GREEN
C***      Red component of the color definition
  CHARACTER*1 RGB_RED
```

RGNRECT Region-rectangle structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
RGNRECT_BLOCK)
C***      Rectangle number from which to start enumerating
  INTEGER*4 RGNRECT_START
C***      Number of rectangles that can be returned
  INTEGER*4 RGNRECT_COUNT
C***      Number of rectangles returned
  INTEGER*4 RGNRECT_RETCOUNT
C***      Direction in which the returned rectangles are to be
ordered
  INTEGER*4 RGNRECT_DIRECTION
```

ROF

Number representation, equivalent to *FIXED*.
(Maps via FIXED)

ROL

Number representation, equivalent to *LONG*.
(Maps via LONG)

SBCDATA

Scroll-bar control data structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
SBCDATA_BLOCK)
 C***      Length of control data in bytes
   INTEGER*4 SBCDATA_LENGTH
 C***      Highlighting code
   INTEGER*4 SBCDATA_HILITE
 C***      First bound of the scroll-bar range
   INTEGER*4 SBCDATA_FIRST
 C***      Last bound of the scroll-bar range
   INTEGER*4 SBCDATA_LAST
 C***      Slider position
   INTEGER*4 SBCDATA_SLIDER
 C***      Number of data items visible
   INTEGER*4 SBCDATA_VISIBLE
 C***      Number of data items available
   INTEGER*4 SBCDATA_TOTAL
```

SEGOFF

2-byte segment offset in bytes.

```
 INTEGER*4
```

SFACTORS

Scaling factors, see DevEscape.
It is represented by the pair of values:

X, Y: INTEGER*4

SHORT

Signed integer in the range –32 768 THROUGH 32 767.

```
 INTEGER*4
```

SIZEF

Size structure.
It is represented by the pair of values:

Width, Height: INTEGER*4

SIZEL

Size structure.
It is represented by the pair of values:

Width, Height: INTEGER*4

SIZEROF

Size structure.
(Maps via SIZEF)

SIZEROL

Size structure.
(Maps via SIZEL)

SMHSTRUCT

Send message hook structure.

```
 C*** DECLAREs for structure components (EQUIVALENCEd onto
SMH_BLOCK)
 C***      Parameter 2
   INTEGER*4 SMH_PARAM2
 C***      Parameter 1
   INTEGER*4 SMH_PARAM1
 C***      Message identity
   INTEGER*4 SMH_MSGID
 C***      Window handle
   INTEGER*4 SMH_HWND
```

STORAGE
Storage that may contain different data types.
(HSTRUCT: Use composition or decomposition functions.)

STR
String with an explicit length count.
As a component of a structure:
(HSTRUCT: Use composition or decomposition functions.)

As a function parameter:
CHARACTER*(length), according to a length parameter.

STRCOND
String with an explicit length count.
(Maps via STR)

STRL
In this language, the datatype maps as follows:

In a data structure:
HSTRUCT (Use composition or decomposition functions.)

As an array function parameter:
An array of HSTRUCTs

As an INPUT or INPUT-OUTPUT function parameter:
A Length/String pair, as shown below

As an OUTPUT or RETURN function parameter:

If there is an 'output length' function parameter 'n':
As STR(n)
Otherwise:
A String/Length pair, as shown below

Length:	INTEGER*4
String:	CHARACTER*(length)

STRLLIST
A list of null-terminated strings (that is, a series of concatenated strings each of which has a *STRL* data type) terminated by an additional null.
(HSTRUCT: Use composition or decomposition functions.)

STR16
String of characters, with an implicit length, in a 16-byte field.

It is represented by the pair of values:
Length: INTEGER*4
String: CHARACTER*16

STR32
String of characters, with an implicit length, in a 32-byte field.

It is represented by the pair of values:
Length: INTEGER*4
String: CHARACTER*32

STR64
String of characters, with an implicit length, in a 64-byte field.

It is represented by the pair of values:
Length: INTEGER*4
String: CHARACTER*64

STR8
String of 8 characters.
CHARACTER*8

SWBLOCK
Switch list block structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
SWBLOCK_BLOCK)
C***     Count of switch list entries
  INTEGER*4               SWBLOCK_COUNT
C***     Switch list entries
  CHARACTER*(SWENTRY_LEN) SWBLOCK_SWENTRY(SWBLOCK_DIM)
```

SWCNTRL Switch-list control block structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
SWCNTRL_BLOCK)
C***      Window handle
  INTEGER*4     SWCNTRL_HWND
C***      Window-handle icon
  INTEGER*4     SWCNTRL_ICON
C***      Program handle
  INTEGER*4     SWCNTRL_HPROG
C***      Process identity
  INTEGER*4     SWCNTRL_PROCESS
C***      Session identity
  INTEGER*4     SWCNTRL_SESSION
C***      Visibility
  INTEGER*4     SWCNTRL_VISIBILITY
C***      Jump indicator
  INTEGER*4     SWCNTRL_JUMP
C***      Switch-list control block title (null terminated)
  INTEGER*4     SWCNTRL_LENGTH
  CHARACTER*61 SWCNTRL_SWTITLE
C***      Reserved
  CHARACTER*1   SWCNTRL_RESERVED
```

SWENTRY Switch list entry structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
SWENTRY_BLOCK)
C***      Switch list entry handle
  INTEGER*4                 SWENTRY_HSWITCH
C***      Switch list control block structure
  CHARACTER*(SWCNTRL_LEN) SWENTRY_SWCNTRL
```

SWP Set Window Position structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
SWP_BLOCK)
C***      Options
  INTEGER*4 SWP_OPTIONS
C***      Window height
  INTEGER*4 SWP_HEIGHT
C***      Window width
  INTEGER*4 SWP_WIDTH
C***      y coordinate of origin
  INTEGER*4 SWP_Y
C***      x coordinate of origin
  INTEGER*4 SWP_X
C***      Window behind which this window is placed
  INTEGER*4 SWP_BEHIND
C***      Window handle
  INTEGER*4 SWP_HWND
```

TID Thread identity.

```
INTEGER*4
```

TIME Time interval in milliseconds.

```
INTEGER*4
```

TRACKINFO Tracking-information structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
TRACKINF_BLOCK)
C***      Border width
  INTEGER*4 TRACKINF_BORDERWIDTH
C***      Border height
  INTEGER*4 TRACKINF_BORDERHEIGHT
C***      Grid width
  INTEGER*4 TRACKINF_GRIDWIDTH
C***      Grid height
  INTEGER*4 TRACKINF_GRIDHEIGHT
C***      Character cell width movement for arrow key
  INTEGER*4 TRACKINF_XKEYBOARD
C***      Character cell height movement for arrow key
  INTEGER*4 TRACKINF_YKEYBOARD
C***      Starting tracking rectangle
  INTEGER*4 TRACKINF_XLEFT
  INTEGER*4 TRACKINF_YBOTTOM
  INTEGER*4 TRACKINF_XRIGHT
  INTEGER*4 TRACKINF_YTOP
C***      Boundary rectangle
  INTEGER*4 TRACKINF_XLEFT_2
  INTEGER*4 TRACKINF_YBOTTOM_2
  INTEGER*4 TRACKINF_XRIGHT_2
  INTEGER*4 TRACKINF_YTOP_2
C***      Minimum tracking size
  INTEGER*4 TRACKINF_X
  INTEGER*4 TRACKINF_Y
C***      Maximum tracking size
  INTEGER*4 TRACKINF_X_2
  INTEGER*4 TRACKINF_Y_2
C***      Tracking options
  INTEGER*4 TRACKINF_OPTIONS
```

UCHAR Unsigned integer in the range 0 through 255.

```
INTEGER*4
```

ULONG Unsigned integer in the range 0 through 2 147 483 647.

```
INTEGER*4
```

USERBUTTON User-button data structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
USERBTN_BLOCK)
C***      Window handle
  INTEGER*4 USERBTN_HWND
C***      Presentation-space handle
  INTEGER*4 USERBTN_HPS
C***      New state of user button
  INTEGER*4 USERBTN_STATE
C***      Old state of user button
  INTEGER*4 USERBTN_OLDSTATE
```

USHORT Unsigned integer in the range 0 through 65 535.

```
INTEGER*4
```

WIDTH4

Width in the range -2 147 483 648 through 2 147 483 647.

```
INTEGER*4
```

WNDPARAMS

Window parameters.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
WNDPARMS_BLOCK)
C***      Window parameter selection
  INTEGER*4 WNDPARMS_STATUS
C***      Length of window text
  INTEGER*4 WNDPARMS_LENGTH
C***      Window text. (STRL->HSTRUCT)
  INTEGER*4 WNDPARMS_TEXT
C***      Length of presentation parameters
  INTEGER*4 WNDPARMS_PRESPARAMSLENGTH
C***      Presentation parameters. (PRESDATA->HSTRUCT)
  INTEGER*4 WNDPARMS_PRESPARAMS
C***      Length of window class specific data
  INTEGER*4 WNDPARMS_CTLDATALENGTH
C***      Window class specific data. (CTLDATA->HSTRUCT)
  INTEGER*4 WNDPARMS_CTLDATA
```

WNDPROC

Window-procedure identifier.

```
INTEGER*4
```

WPOINT

Window point structure (integer).
It is represented by the 4 values:

X:	INTEGER*4
Dummy1:	INTEGER*4
Y:	INTEGER*4
Dummy2:	INTEGER*4

XYWINSIZE

Window position and size structure.

```
C*** DECLAREs for structure components (EQUIVALENCEd onto
XYWINSIZ_BLOCK)
C***      x coordinate of window origin
  INTEGER*4 XYWINSIZ_X
C***      y coordinate of window origin
  INTEGER*4 XYWINSIZ_Y
C***      Window width
  INTEGER*4 XYWINSIZ_WIDTH
C***      Window height
  INTEGER*4 XYWINSIZ_HEIGHT
C***      Window options
  INTEGER*4 XYWINSIZ_WINDOW
```

Chapter 3. Device Function Calls

DevCloseDC - WDCLS

```
CALL WDCLS (HDC, HMF)

C***                    Device-context handle
      INTEGER*4   HDC
C***                    Error indicator/metafile handle (for a metafile device context)
      INTEGER*4   HMF
```

DevEscape - WDESC

```
CALL WDESC (HDC, CODE, INCOUNT, INDATA, OUTCOUNT, OUTDATA, RESULT)

C***                                    Device-context handle
      INTEGER*4             HDC
C***                                    Escape code
      INTEGER*4             CODE
C***                                    Input data count
      INTEGER*4             INCOUNT
C***                                    The input data required for this escape
      CHARACTER*(incount)   INDATA
C***                                    Output data count
      INTEGER*4             OUTCOUNT
C***                                    Output data
      CHARACTER*(outcount)  OUTDATA
C***                                    Implemented/error indicator
      INTEGER*4             RESULT
```

DevOpenDC - WDOPEN

```
CALL WDOPEN (HAB, TYPE, LENGTH, STRING, COUNT, DATA, COMP, HDC)

C***                                    Anchor-block handle
      INTEGER*4               HAB
C***                                    Type of device context
      INTEGER*4               TYPE
C***                                    Device-information token
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                    Number of items
      INTEGER*4               COUNT
C***                                    Open-device-context data area (EQUIVALENCEd onto
                                        DEVSTRUC_BLOCK)
      CHARACTER*(DEVSTRUC_LEN) DATA
C***                                    Compatible-device-context handle
      INTEGER*4               COMP
C***                                    Device-context handle
      INTEGER*4               HDC
```

DevPostDeviceModes - WDPDM

```
      CALL WDPDM (HAB, DRIVERDATA, LENGTH, STRING, LENGTH_2, DEVICENAME, LENGTH_3, STRING_2,
                  OPTIONS, DRIVERCOUNT)

C***                                        Anchor-block handle
      INTEGER*4               HAB
C***                                        Driver data (DRIVDATA->HSTRUCT)
      INTEGER*4               DRIVERDATA
C***                                        This is a string containing the name of the
                                            presentation device driver
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                        Device-type name
      INTEGER*4               LENGTH_2
      CHARACTER*32            DEVICENAME
C***                                        Device name
      INTEGER*4               LENGTH_3
      CHARACTER*(length_3)    STRING_2
C***                                        Dialog options
      INTEGER*4               OPTIONS
C***                                        Size/error indicator
      INTEGER*4               DRIVERCOUNT
```

DevQueryCaps - WDQCAP

```
      CALL WDQCAP (HDC, START, COUNT, ARRAY, SUCCESS)

C***                              Device-context handle
      INTEGER*4   HDC
C***                              First item of information
      INTEGER*4   START
C***                              Count of items of information
      INTEGER*4   COUNT
C***                              Device capabilities
      INTEGER*4   ARRAY(count)
C***                              Success indicator
      INTEGER*4   SUCCESS
```

DevQueryDeviceNames - WDQDN

```
      CALL WDQDN (HAB, LENGTH, STRING, DN, LENGTH_2, DEVICENAME, LENGTH_3, DEVICEDESC, DT,
                  LENGTH_4, DATATYPE, SUCCESS)

C***                                      Anchor-block handle
      INTEGER*4           HAB
C***                                      Fully qualified name of the file containing the
                                          presentation driver
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                      Maximum number of device names and descriptions that
                                          can be returned
      INTEGER*4           DN
C***                                      Device-name array
      INTEGER*4           LENGTH_2(dn)
      CHARACTER*32        DEVICENAME(dn)
C***                                      Device description array
      INTEGER*4           LENGTH_3(dn)
      CHARACTER*64        DEVICEDESC(dn)
C***                                      Maximum number of data types that can be returned
      INTEGER*4           DT
C***                                      Data type array
      INTEGER*4           LENGTH_4(dt)
      CHARACTER*16        DATATYPE(dt)
C***                                      Success indicator
      INTEGER*4           SUCCESS
```

DevQueryHardcopyCaps - WDQHC

```
      CALL WDQHC (HDC, STARTFORM, FORMS, HCINFO, FORMSRETURNED)

C***                                          Device-context handle
      INTEGER*4               HDC
C***                                          Start-forms code
      INTEGER*4               STARTFORM
C***                                          Number of forms to query
      INTEGER*4               FORMS
C***                                          Hardcopy capabilities information (EQUIVALENCEd
                                              onto HCINFO_BLOCK)
      CHARACTER*(HCINFO_LEN)  HCINFO
C***                                          Details of forms
      INTEGER*4               FORMSRETURNED
```

Chapter 4. Graphics Presentation Interface Function Calls

GpiAssociate - GSASS

```
      CALL GSASS (HPS, HDC, SUCCESS)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Device-context handle
      INTEGER*4   HDC
C***                          Success indicator
      INTEGER*4   SUCCESS
```

GpiBeginArea - GSAREA

```
      CALL GSAREA (HPS, OPTIONS)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Area options
      INTEGER*4   OPTIONS
```

GpiBeginElement - GSELEM

```
      CALL GSELEM (HPS, TYPE, LENGTH, STRING)

C***                                  Presentation-space handle
      INTEGER*4           HPS
C***                                  Type to be associated with element
      INTEGER*4           TYPE
C***                                  Description
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
```

GpiBeginPath - GSPATH

```
      CALL GSPATH (HPS, PATH)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Path identifier
      INTEGER*4   PATH
```

GpiBitBlt - GIBB

```
      CALL GIBB (TARGET, SOURCE, COUNT, X, Y, ROP, OPTIONS)

C***                                  Target presentation-space handle
      INTEGER*4   TARGET
C***                                  Source presentation-space handle
      INTEGER*4   SOURCE
C***                                  Point count
      INTEGER*4   COUNT
C***                                  Point array
      INTEGER*4   X(count), Y(count)
C***                                  Mixing function required
      INTEGER*4   ROP
C***                                  Options
      INTEGER*4   OPTIONS
```

GpiBox - GIBOX

```
      CALL GIBOX (HPS, CONTROL, X, Y, HROUND, VROUND)

C***                                Presentation-space handle
      INTEGER*4   HPS
C***                                Outline and fill control
      INTEGER*4   CONTROL
C***                                Corner point
      INTEGER*4   X, Y
C***                                Corner-rounding control
      INTEGER*4   HROUND
C***                                Corner-rounding control
      INTEGER*4   VROUND
```

GpiCallSegmentMatrix - GICALM

```
      CALL GICALM (HPS, SEGMENT, COUNT, ARRAY, OPTIONS)

C***                                        Presentation-space handle
      INTEGER*4                 HPS
C***                                        Identifier of segment to be called
      INTEGER*4                 SEGMENT
C***                                        Number of elements
      INTEGER*4                 COUNT
C***                                        Instance transform matrix (EQUIVALENCEd onto
                                            MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
C***                                        Transformation options
      INTEGER*4                 OPTIONS
```

GpiCharString - GSCHAP

```
      CALL GSCHAP (HPS, COUNT, STRING)

C***                              Presentation-space handle
      INTEGER*4          HPS
C***                              Number of characters
      INTEGER*4          COUNT
C***                              Characters to be drawn
      CHARACTER*(count)  STRING
```

GpiCharStringAt - GICHAR

```
      CALL GICHAR (HPS, X, Y, COUNT, STRING)

C***                              Presentation-space handle
      INTEGER*4          HPS
C***                              Starting position
      INTEGER*4          X, Y
C***                              Number of characters
      INTEGER*4          COUNT
C***                              Characters to be drawn
      CHARACTER*(count)  STRING
```

GpiCharStringPos - GICHPP

```
CALL GICHPP (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, OPTIONS, COUNT, STRING, ADX)

C***                                    Presentation-space handle
      INTEGER*4          HPS
C***                                    Rectangle structure
      INTEGER*4          XLEFT
      INTEGER*4          YBOTTOM
      INTEGER*4          XRIGHT
      INTEGER*4          YTOP
C***                                    Formatting options
      INTEGER*4          OPTIONS
C***                                    Number of characters in the string
      INTEGER*4          COUNT
C***                                    Character string
      CHARACTER*(count)  STRING
C***                                    Increment values
      INTEGER*4          ADX(count)
```

GpiCharStringPosAt - GICHPR

```
CALL GICHPR (HPS, X, Y, XLEFT, YBOTTOM, XRIGHT, YTOP, OPTIONS, COUNT, STRING, ADX)

C***                                    Presentation-space handle
      INTEGER*4          HPS
C***                                    Starting position
      INTEGER*4          X, Y
C***                                    Rectangle structure
      INTEGER*4          XLEFT
      INTEGER*4          YBOTTOM
      INTEGER*4          XRIGHT
      INTEGER*4          YTOP
C***                                    Formatting options
      INTEGER*4          OPTIONS
C***                                    Number of characters in the string
      INTEGER*4          COUNT
C***                                    Character string
      CHARACTER*(count)  STRING
C***                                    Increment values
      INTEGER*4          ADX(count)
```

GpiCloseFigure - GSFCLS

```
CALL GSFCLS (HPS)

C***                Presentation-space handle
      INTEGER*4   HPS
```

GpiCloseSegment - GSSCLS

```
CALL GSSCLS (HPS)

C***                Presentation-space handle
      INTEGER*4   HPS
```

GpiCombineRegion - GSCREG

```
      CALL GSCREG (HPS, DEST, SRC1, SRC2, MODE, COMPLEXITY)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Destination-region handle
      INTEGER*4   DEST
C***                          First-source-region handle
      INTEGER*4   SRC1
C***                          Second-source-region handle
      INTEGER*4   SRC2
C***                          Method of combination
      INTEGER*4   MODE
C***                          Complexity of resulting region/error indicator
      INTEGER*4   COMPLEXITY
```

GpiComment - GSCOMM

```
      CALL GSCOMM (HPS, LENGTH, DATA)

C***                               Presentation-space handle
      INTEGER*4           HPS
C***                               Data length
      INTEGER*4           LENGTH
C***                               Comment data
      CHARACTER*(length)  DATA
```

GpiConvert - GICONV

```
      CALL GICONV (HPS, SRC, TARG, COUNT, X, Y)

C***                                    Presentation-space handle
      INTEGER*4   HPS
C***                                    Source-coordinate space
      INTEGER*4   SRC
C***                                    Target-coordinate space
      INTEGER*4   TARG
C***                                    Point count
      INTEGER*4   COUNT
C***                                    Array of (x,y) coordinate pair structures
      INTEGER*4   X(count), Y(count)
```

GpiCopyMetaFile - GSMCPY

```
      CALL GSMCPY (HMF, NEW)

C***                Source metafile handle
      INTEGER*4   HMF
C***                New metafile handle/error indicator
      INTEGER*4   NEW
```

GpiCorrelateChain - GICORS

```
      CALL GICORS (HPS, TYPE, X, Y, MAXHITS, MAXDEPTH, SEGTAG, NUMHITS)

C***                                        Presentation-space handle
      INTEGER*4   HPS
C***                                        Segment type
      INTEGER*4   TYPE
C***                                        Pick position
      INTEGER*4   X, Y
C***                                        Maximum hits
      INTEGER*4   MAXHITS
C***                                        Number of pairs
      INTEGER*4   MAXDEPTH
C***                                        Segment identifiers and tags
      INTEGER*4   SEGTAG(2*maxdepth*maxhits)
C***                                        Number of hits or error
      INTEGER*4   NUMHITS
```

GpiCorrelateFrom - GICORF

```
      CALL GICORF (HPS, FIRSTSEGMENT, LASTSEGMENT, TYPE, X, Y, MAXHITS, MAXDEPTH, SEGTAG,
                   NUMHITS)

C***                                        Presentation-space handle
      INTEGER*4   HPS
C***                                        Specifies the first segment to be correlated
      INTEGER*4   FIRSTSEGMENT
C***                                        Specifies the last segment to be correlated
      INTEGER*4   LASTSEGMENT
C***                                        The type of segments on which correlation is to
                                            be performed
      INTEGER*4   TYPE
C***                                        Pick position
      INTEGER*4   X, Y
C***                                        Maximum hits
      INTEGER*4   MAXHITS
C***                                        Number of pairs
      INTEGER*4   MAXDEPTH
C***                                        An array of segment and tag identifiers in
                                            alternate elements
      INTEGER*4   SEGTAG(2*maxdepth*maxhits)
C***                                        Number of hits, or error
      INTEGER*4   NUMHITS
```

GpiCorrelateSegment - GICSEG

```
      CALL GICSEG (HPS, SEGMENT, TYPE, X, Y, MAXHITS, MAXDEPTH, SEGTAG, NUMHITS)

C***                                        Presentation-space handle
      INTEGER*4   HPS
C***                                        Identifier of the segment to be correlated; it
                                            must be greater than 0
      INTEGER*4   SEGMENT
C***                                        Type of segments on which correlation is to be
                                            performed
      INTEGER*4   TYPE
C***                                        Position of the center of the pick aperture in
                                            presentation page space
      INTEGER*4   X, Y
C***                                        Maximum hits
      INTEGER*4   MAXHITS
C***                                        Number of segment and tag pairs to be returned
                                            for each hit
      INTEGER*4   MAXDEPTH
C***                                        Array
      INTEGER*4   SEGTAG(maxdepth*maxhits)
C***                                        Number of hits, or error
      INTEGER*4   NUMHITS
```

GpiCreateBitmap - GSCBM

```
      CALL GSCBM (HPS, NEW, OPTIONS, INITDATA, INFOTABLE, HBM)

C***                                        Presentation-space handle
      INTEGER*4               HPS
C***                                        Bit-map information header (EQUIVALENCEd onto
                                            BITINFH_BLOCK)
      CHARACTER*(BITINFH_LEN) NEW
C***                                        Options
      INTEGER*4               OPTIONS
C***                                        Buffer address
      CHARACTER*(n)           INITDATA
C***                                        Bit-map information table (EQUIVALENCEd onto
                                            BITINF_BLOCK)
      CHARACTER*(BITINF_LEN)  INFOTABLE
C***                                        Bit-map handle/error indicator
      INTEGER*4               HBM
```

GpiCreateLogColorTable - GSCLCT

```
      CALL GSCLCT (HPS, OPTIONS, FORMAT, START, COUNT, TABLE)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Options
      INTEGER*4   OPTIONS
C***                          Format of entries in the table
      INTEGER*4   FORMAT
C***                          Starting index
      INTEGER*4   START
C***                          Count of elements in Table
      INTEGER*4   COUNT
C***                          Start of the application data area
      INTEGER*4   TABLE(count)
```

GpiCreateLogFont - GSCRLF

```
CALL GSCRLF (HPS, NAME, LCID, ATTRS, MATCH)

C***                                    Presentation-space handle
      INTEGER*4              HPS
C***                                    Logical font name
      CHARACTER*8            NAME
C***                                    Local identifier
      INTEGER*4              LCID
C***                                    Attributes of font (EQUIVALENCEd onto FATTRS_BLOCK)
      CHARACTER*(FATTRS_LEN) ATTRS
C***                                    Match indicator
      INTEGER*4              MATCH
```

GpiCreatePS - GIPCRT

```
CALL GIPCRT (HAB, HDC, WIDTH, HEIGHT, OPTIONS, HPS)

C***                                Anchor-block handle
      INTEGER*4   HAB
C***                                Device-context handle
      INTEGER*4   HDC
C***                                Presentation-page size
      INTEGER*4   WIDTH, HEIGHT
C***                                Options
      INTEGER*4   OPTIONS
C***                                Presentation-space handle
      INTEGER*4   HPS
```

GpiCreateRegion - GIREG

```
CALL GIREG (HPS, COUNT, XLEFT, YBOTTOM, XRIGHT, YTOP, HRGN)

C***                                Presentation-space handle
      INTEGER*4   HPS
C***                                The number of rectangles
      INTEGER*4   COUNT
C***                                An array of rectangles
      INTEGER*4   XLEFT(count)
      INTEGER*4   YBOTTOM(count)
      INTEGER*4   XRIGHT(count)
      INTEGER*4   YTOP(count)
C***                                Region handle
      INTEGER*4   HRGN
```

GpiDeleteBitmap - GSDBM

```
CALL GSDBM (HBM)

C***                Handle of bit map to be deleted
      INTEGER*4   HBM
```

GpiDeleteElement - GSDLE

```
CALL GSDLE (HPS)

C***                Presentation-space handle
      INTEGER*4   HPS
```

GpiDeleteElementRange - GSDLER

```
CALL GSDLER (HPS, FIRSTELEMENT, LASTELEMENT)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Number of the first element to be deleted
      INTEGER*4   FIRSTELEMENT
C***                          Number of the last element to be deleted
      INTEGER*4   LASTELEMENT
```

GpiDeleteElementsBetweenLabels - GSDELB

```
CALL GSDELB (HPS, FIRSTLABEL, LASTLABEL)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Label marking the start of the elements to be deleted
      INTEGER*4   FIRSTLABEL
C***                          Label marking the end of the elements to be deleted
      INTEGER*4   LASTLABEL
```

GpiDeleteMetaFile - GSMDEL

```
CALL GSMDEL (HMF)

C***                  Metafile handle
      INTEGER*4   HMF
```

GpiDeleteSegment - GSSDEL

```
CALL GSSDEL (HPS, SEGID)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Segment identifier
      INTEGER*4   SEGID
```

GpiDeleteSegments - GSDELS

```
CALL GSDELS (HPS, FIRSTSEGMENT, LASTSEGMENT)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          First identifier in the range; it must be greater than 0
      INTEGER*4   FIRSTSEGMENT
C***                          Last identifier in the range; it must be greater than 0
      INTEGER*4   LASTSEGMENT
```

GpiDeleteSetId - GSRSS

```
CALL GSRSS (HPS, LCID)

C***                 Presentation-space handle
      INTEGER*4   HPS
C***                 Local identifier
      INTEGER*4   LCID
```

GpiDestroyPS - GIPDEL

```
CALL GIPDEL (HPS)

C***                   Presentation-space handle
      INTEGER*4   HPS
```

GpiDestroyRegion - GSDREG

```
CALL GSDREG (HPS, HRGN)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Handle of region to be destroyed
      INTEGER*4   HRGN
```

GpiDrawChain - GSDCHN

```
CALL GSDCHN (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiDrawDynamics - GSDDYN

```
CALL GSDDYN (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiDrawFrom - GSDF

```
CALL GSDF (HPS, FIRSTSEGMENT, LASTSEGMENT)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          First segment to be drawn; it must be greater than zero
      INTEGER*4   FIRSTSEGMENT
C***                          Last segment to be drawn; it must be greater than zero
      INTEGER*4   LASTSEGMENT
```

GpiDrawSegment - GSDSEG

```
CALL GSDSEG (HPS, SEGMENT)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Segment to be drawn; it must be greater than zero
      INTEGER*4   SEGMENT
```

GpiEndArea - GSENDA

```
CALL GSENDA (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiEndElement - GSENDE

```
CALL GSENDE (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiEndPath - GSENDP

```
CALL GSENDP (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiEqualRegion - GSEREG

```
      CALL GSEREG (HPS, SRC1, SRC2)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Handle of first region
      INTEGER*4   SRC1
C***                    Handle of second region
      INTEGER*4   SRC2
```

GpiErase - GSERSE

```
      CALL GSERSE (HPS)

C***                    Presentation-space handle
      INTEGER*4   HPS
```

GpiErrorSegmentData - GSESD

```
      CALL GSESD (HPS, SEGMENT, CONTEXT, OFF)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Segment in which the error occurred
      INTEGER*4   SEGMENT
C***                      Context of the error
      INTEGER*4   CONTEXT
C***                      Position
      INTEGER*4   OFF
```

GpiExcludeClipRectangle - GIECR

```
      CALL GIECR (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Rectangle to be excluded
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiFillPath - GSFILP

```
      CALL GSFILP (HPS, PATH, OPTIONS)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Identifier of path whose interior is to be drawn; it must be 1
      INTEGER*4   PATH
C***                       Fill option
      INTEGER*4   OPTIONS
```

GpiFullArc - GIFARC

```
      CALL GIFARC (HPS, CONTROL, MULTIPLIER)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Interior/outline control
      INTEGER*4   CONTROL
C***                          Multiplier
      INTEGER*4   MULTIPLIER
```

GpiGetData - GSGET

```
      CALL GSGET (HPS, SEGID, OFFSET, FORMAT, LENGTH, DATA)

C***                                  Presentation-space handle
      INTEGER*4           HPS
C***                                  Segment identifier
      INTEGER*4           SEGID
C***                                  Segment offset
      INTEGER*4           OFFSET
C***                                  Coordinate type required
      INTEGER*4           FORMAT
C***                                  Length of data buffer
      INTEGER*4           LENGTH
C***                                  Data buffer
      CHARACTER*(length)  DATA
```

GpiImage - GSIMG

```
      CALL GSIMG (HPS, FORMAT, WIDTH, HEIGHT, LENGTH, DATA)

C***                                        Presentation-space handle
      INTEGER*4           HPS
C***                                        Format of image data
      INTEGER*4           FORMAT
C***                                        Size of image area (in pels)
      INTEGER*4           WIDTH, HEIGHT
C***                                        Length in bytes of image data
      INTEGER*4           LENGTH
C***                                        Image data
      CHARACTER*(length)  DATA
```

GpiIntersectClipRectangle - GIICR

```
      CALL GIICR (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, COMPLEXITY)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Rectangle, the coordinates of which are world coordinates
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Complexity of clipping/error indicator
      INTEGER*4   COMPLEXITY
```

GpiLabel - GSINLB

```
      CALL GSINLB (HPS, LABEL)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Required label
      INTEGER*4   LABEL
```

GpiLine - GILINE

```
      CALL GILINE (HPS, X, Y)

C***                 Presentation-space handle
      INTEGER*4   HPS
C***                 End point of the line
      INTEGER*4   X, Y
```

GpiLoadBitmap - GSLBM

```
      CALL GSLBM (HPS, RESOURCE, BITMAP, WIDTH, HEIGHT, HBM)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Resource identity containing the bit map
      INTEGER*4   RESOURCE
C***                          ID of the bit map within the resource file
      INTEGER*4   BITMAP
C***                          Width of the bit map in pels
      INTEGER*4   WIDTH
C***                          Height of the bit map in pels
      INTEGER*4   HEIGHT
C***                          Bit-map handle
      INTEGER*4   HBM
```

GpiLoadFonts - GSFLO

```
      CALL GSFLO (HAB, LENGTH, STRING, SUCCESS)

C***                                Anchor-block handle
      INTEGER*4           HAB
C***                                Filename
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                Success indicator
      INTEGER*4           SUCCESS
```

GpiLoadMetaFile - GSMLOD

```
      CALL GSMLOD (HAB, LENGTH, STRING, HMF)

C***                                Anchor-block handle
      INTEGER*4           HAB
C***                                Filename
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                Metafile handle or error
      INTEGER*4           HMF
```

GpiMarker - GIMARK

```
      CALL GIMARK (HPS, X, Y)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Position of the marker
      INTEGER*4   X, Y
```

GpiModifyPath - GSMODP

```
      CALL GSMODP (HPS, PATH, MODE)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Path identifier
      INTEGER*4   PATH
C***                     Modification required
      INTEGER*4   MODE
```

GpiMove - GIMOVE

```
      CALL GIMOVE (HPS, X, Y)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Position to which to move
      INTEGER*4   X, Y
```

GpiOffsetClipRegion - GIOCR

```
      CALL GIOCR (HPS, X, Y, COMPLEXITY)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Displacement
      INTEGER*4   X, Y
C***                        Complexity of clipping/error indicator
      INTEGER*4   COMPLEXITY
```

GpiOffsetElementPointer - GSOEP

```
      CALL GSOEP (HPS, OFFSET)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Offset to be added to the element pointer
      INTEGER*4   OFFSET
```

GpiOffsetRegion - GIOREG

```
      CALL GIOREG (HPS, HRGN, X, Y)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Region handle to be moved
      INTEGER*4   HRGN
C***                  Offset to be added to the region boundary
      INTEGER*4   X, Y
```

GpiOpenSegment - GSSEG

```
      CALL GSSEG (HPS, SEGMENT)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Segment identifier
      INTEGER*4   SEGMENT
```

GpiOutlinePath - GSOUTP

```
      CALL GSOUTP (HPS, PATH, OPTIONS)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Identifier of path to be outlined; it must be 1
      INTEGER*4   PATH
C***                     Options
      INTEGER*4   OPTIONS
```

GpiPaintRegion - GSPREG

```
      CALL GSPREG (HPS, HRGN)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Region handle
      INTEGER*4   HRGN
```

GpiPartialArc - GIPARC

```
      CALL GIPARC (HPS, X, Y, MULTIPLIER, STARTANGLE, SWEEPANGLE)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Center point
      INTEGER*4   X, Y
C***                          Multiplier
      INTEGER*4   MULTIPLIER
C***                          Start angle in degrees
      INTEGER*4   STARTANGLE
C***                          Sweep angle in degrees
      INTEGER*4   SWEEPANGLE
```

GpiPlayMetaFile - GSLOAD

```
      CALL GSLOAD (HPS, HMF, COUNT1, OPTARRAY, SEGCOUNT, COUNT2, LENGTH, STRING)

C***                                      Presentation-space handle
      INTEGER*4           HPS
C***                                      Metafile handle
      INTEGER*4           HMF
C***                                      Count of elements in Optarray
      INTEGER*4           COUNT1
C***                                      Array of options for playing
      INTEGER*4           OPTARRAY(count1)
C***                                      Reserved
      INTEGER*4           SEGCOUNT
C***                                      Count of bytes in desc
      INTEGER*4           COUNT2
C***                                      Descriptive record (LENGTH is result string
                                          length)
      INTEGER*4           LENGTH
      CHARACTER*(count2)  STRING
```

GpiPointArc - GIGARC

```
      CALL GIGARC (HPS, X, Y)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Intermediate and end points
      INTEGER*4   X(2), Y(2)
```

GpiPolyFillet - GIPFLT

```
      CALL GIPFLT (HPS, COUNT, X, Y)

C***                                Presentation-space handle
      INTEGER*4   HPS
C***                                Number of points
      INTEGER*4   COUNT
C***                                Array of points
      INTEGER*4   X(count), Y(count)
```

GpiPolyFilletSharp - GIPFLS

```
      CALL GIPFLS (HPS, COUNT, X, Y, SHARPNESS)

C***                                  Presentation-space handle
      INTEGER*4   HPS
C***                                  Count of points
      INTEGER*4   COUNT
C***                                  An array of points
      INTEGER*4   X(count), Y(count)
C***                                  Array of sharpness values
      INTEGER*4   SHARPNESS(count/2)
```

GpiPolyLine - GIPLNE

```
      CALL GIPLNE (HPS, COUNT, X, Y)

C***                                  Presentation-space handle
      INTEGER*4   HPS
C***                                  Number of points
      INTEGER*4   COUNT
C***                                  Array of points
      INTEGER*4   X(count), Y(count)
```

GpiPolyMarker - GIMRKS

```
      CALL GIMRKS (HPS, COUNT, X, Y)

C***                                  Presentation-space handle
      INTEGER*4   HPS
C***                                  Number of points
      INTEGER*4   COUNT
C***                                  Array of points
      INTEGER*4   X(count), Y(count)
```

GpiPolySpline - GIPSPL

```
      CALL GIPSPL (HPS, COUNT, X, Y)

C***                                  Presentation-space handle
      INTEGER*4   HPS
C***                                  Count of points
      INTEGER*4   COUNT
C***                                  An array of points
      INTEGER*4   X(count), Y(count)
```

GpiPop - GSPOP

```
      CALL GSPOP (HPS, COUNT)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Number of attributes to be restored
      INTEGER*4   COUNT
```

GpiPtInRegion - GIPIR

```
      CALL GIPIR (HPS, HRGN, X, Y, INSIDE)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Region handle
      INTEGER*4   HRGN
C***                    Point to be checked
      INTEGER*4   X, Y
C***                    Inside/error indicator
      INTEGER*4   INSIDE
```

GpiPtVisible - GIPVIS

```
      CALL GIPVIS (HPS, X, Y, VISIBILITY)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Point to be checked
      INTEGER*4   X, Y
C***                          Visibility indicator
      INTEGER*4   VISIBILITY
```

GpiPutData - GSPUT

```
      CALL GSPUT (HPS, FORMAT, LENGTH, DATA)

C***                              Presentation-space handle
      INTEGER*4           HPS
C***                              Coordinate type used
      INTEGER*4           FORMAT
C***                              Length of graphic data
      INTEGER*4           LENGTH
C***                              Orders to be copied
      CHARACTER*(length)  DATA
```

GpiQueryArcParams - GIQAP

```
      CALL GIQAP (HPS, ARCPARAMS)

C***                                    Presentation-space handle
      INTEGER*4                HPS
C***                                    Arc parameters (EQUIVALENCEd onto ARCPARML_BLOCK)
      CHARACTER*(ARCPARML_LEN)  ARCPARAMS
```

GpiQueryAttrMode - GSQAM

```
      CALL GSQAM (HPS, MODE)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Current attribute mode
      INTEGER*4   MODE
```

GpiQueryAttrs - GIQATR

```
      CALL GIQATR (HPS, PRIMTYPE, ATTRMASK, ATTRS, DEFMASK)

C***                                  Presentation-space handle
      INTEGER*4              HPS
C***                                  Primitive type
      INTEGER*4              PRIMTYPE
C***                                  Attributes mask
      INTEGER*4              ATTRMASK
C***                                  Attributes (EQUIVALENCEd onto BUNDLE_BLOCK)
      CHARACTER*(BUNDLE_LEN)  ATTRS
C***                                  Defaults mask
      INTEGER*4              DEFMASK
```

GpiQueryBackColor - GSQBCO

```
      CALL GSQBCO (HPS, COLOR)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Background color
      INTEGER*4   COLOR
```

GpiQueryBackMix - GSQBMX

```
CALL GSQBMX (HPS, MIXMODE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Background mix
      INTEGER*4   MIXMODE
```

GpiQueryBitmapBits - GSQBB

```
CALL GSQBB (HPS, SCANSTART, SCANS, BUFFER, INFOTABLE, SCANSRETURNED)

C***                                        Presentation-space handle
      INTEGER*4              HPS
C***                                        Starting line number
      INTEGER*4              SCANSTART
C***                                        Number of scan lines to be returned
      INTEGER*4              SCANS
C***                                        Data area
      CHARACTER*(n)          BUFFER
C***                                        Bit-map information table (EQUIVALENCEd onto
                                            BITINF_BLOCK)
      CHARACTER*(BITINF_LEN) INFOTABLE
C***                                        Number of scan lines actually returned
      INTEGER*4              SCANSRETURNED
```

GpiQueryBitmapDimension - GSQBTD

```
CALL GSQBTD (HBM, WIDTH, HEIGHT)

C***                           Bit-map handle
      INTEGER*4   HBM
C***                           Size of bit map
      INTEGER*4   WIDTH, HEIGHT
```

GpiQueryBitmapHandle - GSQBTH

```
CALL GSQBTH (HPS, LCID, HBM)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Local identifier
      INTEGER*4   LCID
C***                Bit-map handle
      INTEGER*4   HBM
```

GpiQueryBitmapParameters - GSQBTP

```
CALL GSQBTP (HBM, DATA)

C***                                Bit-map handle
      INTEGER*4              HBM
C***                                Bit-map information header (EQUIVALENCEd onto
                                    BITINFH_BLOCK)
      CHARACTER*(BITINFH_LEN) DATA
```

GpiQueryBoundaryData - GIQBD

```
CALL GIQBD (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Boundary data
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiQueryCharAngle - GIQCA

```
      CALL GIQCA (HPS, X, Y)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Baseline angle
      INTEGER*4   X, Y
```

GpiQueryCharBox - GIQCB

```
      CALL GIQCB (HPS, WIDTH, HEIGHT)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Character-box size
      INTEGER*4   WIDTH, HEIGHT
```

GpiQueryCharDirection - GSQCD

```
      CALL GSQCD (HPS, DIRECTION)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Character direction
      INTEGER*4   DIRECTION
```

GpiQueryCharMode - GSQCM

```
      CALL GSQCM (HPS, MODE)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Character mode
      INTEGER*4   MODE
```

GpiQueryCharSet - GSQCS

```
      CALL GSQCS (HPS, LCID)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Character-set local identifier
      INTEGER*4   LCID
```

GpiQueryCharShear - GIQCH

```
      CALL GIQCH (HPS, X, Y)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Character shear
      INTEGER*4   X, Y
```

GpiQueryCharStringPos - GIQCPO

```
CALL GIQCPO (HPS, OPTIONS, COUNT, STRING, XINCREMENTS, X, Y)

C***                                          Presentation-space handle
      INTEGER*4          HPS
C***                                          Option flag
      INTEGER*4          OPTIONS
C***                                          Length of the string
      INTEGER*4          COUNT
C***                                          The character string to be examined
      CHARACTER*(count)  STRING
C***                                          Vector of x increment values
      INTEGER*4          XINCREMENTS(count)
C***                                          Array of points
      INTEGER*4          X(count+1), Y(count+1)
```

GpiQueryCharStringPosAt - GIQCRO

```
CALL GIQCRO (HPS, X, Y, OPTIONS, COUNT, STRING, XINCREMENTS, X_2, Y_2)

C***                                              Presentation-space handle
      INTEGER*4          HPS
C***                                              Starting position
      INTEGER*4          X, Y
C***                                              Option flags
      INTEGER*4          OPTIONS
C***                                              Length of the string
      INTEGER*4          COUNT
C***                                              Character string to be examined
      CHARACTER*(count)  STRING
C***                                              Vector of x increment values
      INTEGER*4          XINCREMENTS(count)
C***                                              Array of points, in which the positions
                                                  of each character in world coordinates
                                                  are returned
      INTEGER*4          X_2(count+1), Y_2(count+1)
```

GpiQueryClipBox - GIQCLB

```
CALL GIQCLB (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, COMPLEXITY)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Bounding rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Complexity/error indicator
      INTEGER*4   COMPLEXITY
```

GpiQueryClipRegion - GSQCLR

```
CALL GSQCLR (HPS, HRGN)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Clip-region handle (if any)
      INTEGER*4   HRGN
```

GpiQueryColor - GSQCOL

```
CALL GSQCOL (HPS, COLOR)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Color attribute
      INTEGER*4   COLOR
```

GpiQueryColorData - GSQCOD

```
CALL GSQCOD (HPS, COUNT, ARRAY)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Number of elements
      INTEGER*4   COUNT
C***                              Array
      INTEGER*4   ARRAY(count)
```

GpiQueryColorIndex - GSQCOI

```
CALL GSQCOI (HPS, OPTIONS, RGBCOLOR, INDEX)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Options
      INTEGER*4   OPTIONS
C***                        Specifies a color in RGB terms
      INTEGER*4   RGBCOLOR
C***                        Color index providing closest match to the specified color
      INTEGER*4   INDEX
```

GpiQueryCp - GSQCPG

```
CALL GSQCPG (HPS, CODEPAGE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Code page
      INTEGER*4   CODEPAGE
```

GpiQueryCurrentPosition - GIQCP

```
CALL GIQCP (HPS, X, Y)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Current position
      INTEGER*4   X, Y
```

GpiQueryDefArcParams - GIQDAP

```
CALL GIQDAP (HPS, ARCPARAMS)

C***                                      Presentation-space handle
      INTEGER*4                 HPS
C***                                      Default arc parameters (EQUIVALENCEd onto
                                          ARCPARML_BLOCK)
      CHARACTER*(ARCPARML_LEN)  ARCPARAMS
```

GpiQueryDefAttrs - GIQDAT

```
CALL GIQDAT (HPS, PRIMTYPE, ATTRMASK, ATTRS)

C***                                    Presentation-space handle
      INTEGER*4              HPS
C***                                    Primitive type
      INTEGER*4              PRIMTYPE
C***                                    Attributes mask
      INTEGER*4              ATTRMASK
C***                                    Attributes (EQUIVALENCEd onto BUNDLE_BLOCK)
      CHARACTER*(BUNDLE_LEN) ATTRS
```

GpiQueryDefaultViewMatrix - GIQDVM

```
CALL GIQDVM (HPS, COUNT, ARRAY)

C***                                    Presentation-space handle
      INTEGER*4                HPS
C***                                    Number of elements
      INTEGER*4                COUNT
C***                                    Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN) ARRAY
```

GpiQueryDefCharBox - GIQCEL

```
CALL GIQCEL (HPS, WIDTH, HEIGHT)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            Default character-box size
      INTEGER*4   WIDTH, HEIGHT
```

GpiQueryDefTag - GSQDTA

```
CALL GSQDTA (HPS, TAG)

C***                 Presentation-space handle
      INTEGER*4   HPS
C***                 Default tag identifier
      INTEGER*4   TAG
```

GpiQueryDefViewingLimits - GIQDVL

```
CALL GIQDVL (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Default viewing limits
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiQueryDevice - GSQD

```
CALL GSQD (HPS, HDC)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Device-context handle
      INTEGER*4   HDC
```

GpiQueryDeviceBitmapFormats - GSQDBF

```
      CALL GSQDBF (HPS, COUNT, ARRAY)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Number of elements
      INTEGER*4   COUNT
C***                              Data array
      INTEGER*4   ARRAY(count)
```

GpiQueryDrawControl - GSQDC

```
      CALL GSQDC (HPS, CONTROL, VALUE)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Control whose value is to be returned
      INTEGER*4   CONTROL
C***                      Value of the control
      INTEGER*4   VALUE
```

GpiQueryDrawingMode - GSQDM

```
      CALL GSQDM (HPS, MODE)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Drawing mode
      INTEGER*4   MODE
```

GpiQueryEditMode - GSQEDM

```
      CALL GSQEDM (HPS, MODE)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Current editing mode
      INTEGER*4   MODE
```

GpiQueryElementPointer - GSQEP

```
      CALL GSQEP (HPS, ELEMENT)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Current element pointer
      INTEGER*4   ELEMENT
```

GpiQueryElementType - GSQETS

```
      CALL GSQETS (HPS, TYPE, LENGTH, LENGTH_2, STRING, REQLENGTH)

C***                                  Presentation-space handle
      INTEGER*4           HPS
C***                                  Element type
      INTEGER*4           TYPE
C***                                  Data length
      INTEGER*4           LENGTH
C***                                  Description of data buffer (LENGTH_2 is result string
                                      length)
      INTEGER*4           LENGTH_2
      CHARACTER*(length)  STRING
C***                                  Size of the data required to hold the element content
      INTEGER*4           REQLENGTH
```

GpiQueryFontFileDescriptions - GSQFD

```
CALL GSQFD (HAB, LENGTH, STRING, COUNT, NAMES, REMFONTS)

C***                                        Anchor-block handle
      INTEGER*4               HAB
C***                                        Fully qualified filename
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                        Maximum number of family and facename pairs to be
                                            returned
      INTEGER*4               COUNT
C***                                        Array of font file descriptors (EQUIVALENCEd onto
                                            FFDESCS_BLOCK)
      CHARACTER*(FFDESCS_LEN) NAMES(count)
C***                                        Returns
      INTEGER*4               REMFONTS
```

GpiQueryFontMetrics - GSQFM

```
CALL GSQFM (HPS, METRICSLENGTH, METRICS)

C***                                        Presentation-space handle
      INTEGER*4               HPS
C***                                        Length of metrics
      INTEGER*4               METRICSLENGTH
C***                                        Metrics of font (EQUIVALENCEd onto FMETRS_BLOCK)
      CHARACTER*(FMETRS_LEN)  METRICS
```

GpiQueryFonts - GSQF

```
CALL GSQF (HPS, OPTIONS, LENGTH, STRING, REQFONTS, METRICSLENGTH, METRICS, REMFONTS)

C***                                        Presentation-space handle
      INTEGER*4               HPS
C***                                        Enumeration options
      INTEGER*4               OPTIONS
C***                                        Facename of fonts
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                        Count of fonts
      INTEGER*4               REQFONTS
C***                                        Length of metrics
      INTEGER*4               METRICSLENGTH
C***                                        Metrics of font (EQUIVALENCEd onto
                                            FMETRS_BLOCK)
      CHARACTER*(FMETRS_LEN)  METRICS(reqfonts)
C***                                        Count of fonts not returned
      INTEGER*4               REMFONTS
```

GpiQueryGraphicsField - GIQFLD

```
CALL GIQFLD (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Graphics field
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiQueryInitialSegmentAttrs - GSQATI

```
CALL GSQATI (HPS, ATTRIBUTE, VALUE)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Attribute to be queried
      INTEGER*4   ATTRIBUTE
C***                          Current initial attribute value
      INTEGER*4   VALUE
```

GpiQueryKerningPairs - GSQKPR

```
CALL GSQKPR (HPS, COUNT, DATA, RETURNED)

C***                                          Presentation-space handle
      INTEGER*4                 HPS
C***                                          The number of elements in Data
      INTEGER*4                 COUNT
C***                                          Kerning pairs (EQUIVALENCEd onto KERNPAIR_BLOCK)
      CHARACTER*(KERNPAIR_LEN)  DATA(count)
C***                                          Number returned/error indicator
      INTEGER*4                 RETURNED
```

GpiQueryLineEnd - GSQLE

```
CALL GSQLE (HPS, LINEEND)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Line end
      INTEGER*4   LINEEND
```

GpiQueryLineJoin - GSQLJ

```
CALL GSQLJ (HPS, LINEJOIN)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Line join
      INTEGER*4   LINEJOIN
```

GpiQueryLineType - GSQLT

```
CALL GSQLT (HPS, LINETYPE)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Line type
      INTEGER*4   LINETYPE
```

GpiQueryLineWidth - GIQLW

```
CALL GIQLW (HPS, LINEWIDTH)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Line width
      INTEGER*4   LINEWIDTH
```

GpiQueryLineWidthGeom - GIQGLW

```
CALL GIQGLW (HPS, LINEWIDTH)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Geometric line width
      INTEGER*4   LINEWIDTH
```

GpiQueryLogColorTable - GSQLCT

```
CALL GSQLCT (HPS, OPTIONS, START, COUNT, ARRAY, RETCOUNT)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Specifies options
      INTEGER*4   OPTIONS
C***                              The starting index for which data is to be returned
      INTEGER*4   START
C***                              Count of elements
      INTEGER*4   COUNT
C***                              An array in which the information is returned
      INTEGER*4   ARRAY(count)
C***                              Number of elements returned/error indicator
      INTEGER*4   RETCOUNT
```

GpiQueryMarker - GSQMS

```
CALL GSQMS (HPS, SYMBOL)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Marker symbol
      INTEGER*4   SYMBOL
```

GpiQueryMarkerBox - GIQMB

```
CALL GIQMB (HPS, WIDTH, HEIGHT)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Size of marker box
      INTEGER*4   WIDTH, HEIGHT
```

GpiQueryMarkerSet - GSQMSS

```
CALL GSQMSS (HPS, SET)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Marker set local identifier
      INTEGER*4   SET
```

GpiQueryMetaFileBits - GSMGB

```
CALL GSMGB (HMF, OFFSET, LENGTH, DATA)

C***                             Memory-metafile handle
      INTEGER*4           HMF
C***                             Byte offset
      INTEGER*4           OFFSET
C***                             Length in bytes of the metafile data to copy
      INTEGER*4           LENGTH
C***                             Metafile data
      CHARACTER*(length)  DATA
```

GpiQueryMetaFileLength - GSQMFL

```
CALL GSQMFL (HMF, LENGTH)

C***                     Memory-metafile handle
      INTEGER*4   HMF
C***                     Total length of the metafile
      INTEGER*4   LENGTH
```

GpiQueryMix - GSQMIX

```
      CALL GSQMIX (HPS, MIXMODE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Mix mode
      INTEGER*4   MIXMODE
```

GpiQueryModelTransformMatrix - GIQCTM

```
      CALL GIQCTM (HPS, COUNT, ARRAY)

C***                                    Presentation-space handle
      INTEGER*4                 HPS
C***                                    Number of elements
      INTEGER*4                 COUNT
C***                                    Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
```

GpiQueryNearestColor - GSQNC

```
      CALL GSQNC (HPS, OPTIONS, RGBIN, RGBOUT)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Options
      INTEGER*4   OPTIONS
C***                        Required color
      INTEGER*4   RGBIN
C***                        Nearest available color to the one specified
      INTEGER*4   RGBOUT
```

GpiQueryNumberSetIds - GSQNSS

```
      CALL GSQNSS (HPS, COUNT)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Number of lcids
      INTEGER*4   COUNT
```

GpiQueryPageViewport - GIQPV

```
      CALL GIQPV (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Page viewport
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiQueryPattern - GSQPAT

```
      CALL GSQPAT (HPS, PATTERNSYMBOL)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Pattern symbol
      INTEGER*4   PATTERNSYMBOL
```

GpiQueryPatternRefPoint - GIQPRP

```
      CALL GIQPRP (HPS, X, Y)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Pattern reference point
      INTEGER*4   X, Y
```

GpiQueryPatternSet - GSQPTS

```
      CALL GSQPTS (HPS, SET)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Pattern-set local identifier
      INTEGER*4   SET
```

GpiQueryPel - GIQPEL

```
      CALL GIQPEL (HPS, X, Y, COLOR)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Position in world coordinates
      INTEGER*4   X, Y
C***                    Color index of the pel
      INTEGER*4   COLOR
```

GpiQueryPickAperturePosition - GIQPAP

```
      CALL GIQPAP (HPS, X, Y)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Pick aperture position
      INTEGER*4   X, Y
```

GpiQueryPickApertureSize - GIQPAS

```
      CALL GIQPAS (HPS, WIDTH, HEIGHT)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            Pick aperture size
      INTEGER*4   WIDTH, HEIGHT
```

GpiQueryPS - GIPQRY

```
      CALL GIPQRY (HPS, WIDTH, HEIGHT, OPTIONS)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            Presentation-page size
      INTEGER*4   WIDTH, HEIGHT
C***                            Presentation-space options
      INTEGER*4   OPTIONS
```

GpiQueryRealColors - GSQRC

```
      CALL GSQRC (HPS, OPTIONS, START, COUNT, COLORS, RETCOUNT)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Options
      INTEGER*4   OPTIONS
C***                          Ordinal number of the first color required
      INTEGER*4   START
C***                          Maximum number of elements
      INTEGER*4   COUNT
C***                          Array in which the information is returned
      INTEGER*4   COLORS(count)
C***                          Number of elements returned
      INTEGER*4   RETCOUNT
```

GpiQueryRegionBox - GIQRB

```
      CALL GIQRB (HPS, HRGN, XLEFT, YBOTTOM, XRIGHT, YTOP, COMPLEXITY)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Region handle
      INTEGER*4   HRGN
C***                        Bounding rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Complexity of region/error indicator
      INTEGER*4   COMPLEXITY
```

GpiQueryRegionRects - GIQRR

```
      CALL GIQRR (HPS, HRGN, XLEFT, YBOTTOM, XRIGHT, YTOP, CONTROL, XLEFT_2, YBOTTOM_2,
                  XRIGHT_2, YTOP_2)

C***                                                Presentation-space handle
      INTEGER*4                 HPS
C***                                                Region handle
      INTEGER*4                 HRGN
C***                                                Bounding rectangle
      INTEGER*4                 XLEFT
      INTEGER*4                 YBOTTOM
      INTEGER*4                 XRIGHT
      INTEGER*4                 YTOP
C***                                                Processing-control structure
                                                    (EQUIVALENCEd onto RGNRECT_BLOCK)
      CHARACTER*(RGNRECT_LEN)   CONTROL
C***                                                Array of rectangle structures, in
                                                    which the rectangles are returned
      INTEGER*4                 XLEFT_2(rgnrect.count)
      INTEGER*4                 YBOTTOM_2(rgnrect.count)
      INTEGER*4                 XRIGHT_2(rgnrect.count)
      INTEGER*4                 YTOP_2(rgnrect.count)
```

GpiQueryRGBColor - GSQRGB

```
CALL GSQRGB (HPS, OPTIONS, COLORINDEX, RGBCOLOR)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Options
      INTEGER*4   OPTIONS
C***                          Color index
      INTEGER*4   COLORINDEX
C***                          RGB color providing closest match to the specified color index
      INTEGER*4   RGBCOLOR
```

GpiQuerySegmentAttrs - GSQATS

```
CALL GSQATS (HPS, SEGID, ATTRIBUTE, VALUE)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Segment identifier; must be greater than 0
      INTEGER*4   SEGID
C***                          Attribute to be queried
      INTEGER*4   ATTRIBUTE
C***                          Current attribute value
      INTEGER*4   VALUE
```

GpiQuerySegmentNames - GSQNAM

```
CALL GSQNAM (HPS, FIRSTSEGID, LASTSEGID, MAX, SEGIDS, RETCOUNT)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            First segment in the range; must be greater than 0
      INTEGER*4   FIRSTSEGID
C***                            Last segment in the range; must be greater than 0
      INTEGER*4   LASTSEGID
C***                            Maximum number
      INTEGER*4   MAX
C***                            Array in which the required identifiers are returned
      INTEGER*4   SEGIDS(max)
C***                            Number of identifiers returned
      INTEGER*4   RETCOUNT
```

GpiQuerySegmentPriority - GSQPRI

```
CALL GSQPRI (HPS, REFSEGID, ORDER, SEGID)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Reference segment identifier
      INTEGER*4   REFSEGID
C***                          Segment higher or lower
      INTEGER*4   ORDER
C***                          Segment identifier
      INTEGER*4   SEGID
```

GpiQuerySegmentTransformMatrix - GIQTFM

```
      CALL GIQTFM (HPS, SEGID, COUNT, ARRAY)

C***                                        Presentation-space handle
      INTEGER*4                  HPS
C***                                        Segment identifier
      INTEGER*4                  SEGID
C***                                        Number of elements
      INTEGER*4                  COUNT
C***                                        Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)   ARRAY
```

GpiQuerySetIds - GSQSS

```
      CALL GSQSS (HPS, COUNT, TYPES, NAMES, LCIDS)

C***                              Presentation-space handle
      INTEGER*4    HPS
C***                              The number of objects to be queried
      INTEGER*4    COUNT
C***                              Object types
      INTEGER*4    TYPES(count)
C***                              Font names
      CHARACTER*8  NAMES(count)
C***                              Local identifiers
      INTEGER*4    LCIDS(count)
```

GpiQueryStopDraw - GSQSDW

```
      CALL GSQSDW (HPS, VALUE)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Stop draw condition indicator
      INTEGER*4   VALUE
```

GpiQueryTag - GSQTAG

```
      CALL GSQTAG (HPS, TAG)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Tag identifier
      INTEGER*4   TAG
```

GpiQueryTextBox - GIQTB

```
      CALL GIQTB (HPS, COUNT1, STRING, COUNT2, X, Y)

C***                                          Presentation-space handle
      INTEGER*4           HPS
C***                                          Number of characters
      INTEGER*4           COUNT1
C***                                          The character string
      CHARACTER*(count1)  STRING
C***                                          Number of points
      INTEGER*4           COUNT2
C***                                          List of points
      INTEGER*4           X(count2), Y(count2)
```

GpiQueryViewingLimits - GIQSVL

```
CALL GIQSVL (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Viewing limits
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiQueryViewingTransformMatrix - GIQVTM

```
CALL GIQVTM (HPS, COUNT, ARRAY)

C***                                    Presentation-space handle
      INTEGER*4                 HPS
C***                                    Number of elements
      INTEGER*4                 COUNT
C***                                    Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
```

GpiQueryWidthTable - GSQFWT

```
CALL GSQFWT (HPS, FIRSTCHAR, COUNT, DATA)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            Codepoint of first character
      INTEGER*4   FIRSTCHAR
C***                            Count of elements in Data
      INTEGER*4   COUNT
C***                            Array of width values
      INTEGER*4   DATA(count)
```

GpiRealizeColorTable - GSRCT

```
CALL GSRCT (HPS)

C***                Presentation-space handle
      INTEGER*4   HPS
```

GpiRectInRegion - GIRIR

```
CALL GIRIR (HPS, HRGN, XLEFT, YBOTTOM, XRIGHT, YTOP, INSIDE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Region handle
      INTEGER*4   HRGN
C***                        Test rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Inside/error indicator
      INTEGER*4   INSIDE
```

GpiRectVisible - GIRVIS

```
      CALL GIRVIS (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, VISIBILITY)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Test rectangle, in world coordinates
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Visibility indicator
      INTEGER*4   VISIBILITY
```

GpiRemoveDynamics - GSRDYN

```
      CALL GSRDYN (HPS, FIRSTSEGID, LASTSEGID)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          First segment in the section
      INTEGER*4   FIRSTSEGID
C***                          Last segment in the section
      INTEGER*4   LASTSEGID
```

GpiResetBoundaryData - GSRBD

```
      CALL GSRBD (HPS)

C***                Presentation-space handle
      INTEGER*4   HPS
```

GpiResetPS - GSRPS

```
      CALL GSRPS (HPS, OPTIONS)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Reset option
      INTEGER*4   OPTIONS
```

GpiRestorePS - GSRSPS

```
      CALL GSRSPS (HPS, PSID)

C***                   Presentation-space handle
      INTEGER*4   HPS
C***                   Identifier of the saved presentation space that is to be restored
      INTEGER*4   PSID
```

GpiRotate - GIROT

```
      CALL GIROT (HPS, ARRAY, OPTIONS, ANGLE, X, Y)

C***                                 Presentation-space handle
      INTEGER*4                HPS
C***                                 Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN) ARRAY
C***                                 Transform options
      INTEGER*4                OPTIONS
C***                                 Rotation angle
      INTEGER*4                ANGLE
C***                                 Center of rotation
      INTEGER*4                X, Y
```

GpiSaveMetaFile - GSMSAV

```
CALL GSMSAV (HMF, LENGTH, STRING)

C***                                Metafile handle
      INTEGER*4           HMF
C***                                Filename
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
```

GpiSavePS - GSSPS

```
CALL GSSPS (HPS, PSID)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Identifier of saved presentation space
      INTEGER*4   PSID
```

GpiScale - GISCAL

```
CALL GISCAL (HPS, ARRAY, OPTIONS, SCALE, X, Y)

C***                                    Presentation-space handle
      INTEGER*4                 HPS
C***                                    Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
C***                                    Transform options
      INTEGER*4                 OPTIONS
C***                                    Scale factors
      INTEGER*4                 SCALE(2)
C***                                    Center of scale
      INTEGER*4                 X, Y
```

GpiSetArcParams - GISAP

```
CALL GISAP (HPS, ARCPARAMS)

C***                                    Presentation-space handle
      INTEGER*4                 HPS
C***                                    Arc parameters (EQUIVALENCEd onto ARCPARML_BLOCK)
      CHARACTER*(ARCPARML_LEN)  ARCPARAMS
```

GpiSetAttrMode - GSAM

```
CALL GSAM (HPS, MODE)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Attribute mode
      INTEGER*4   MODE
```

GpiSetAttrs - GIATR

```
CALL GIATR (HPS, PRIMTYPE, ATTRMASK, DEFMASK, ATTRS)

C***                                    Presentation-space handle
      INTEGER*4                 HPS
C***                                    Primitive type
      INTEGER*4                 PRIMTYPE
C***                                    Attributes mask
      INTEGER*4                 ATTRMASK
C***                                    Defaults mask
      INTEGER*4                 DEFMASK
C***                                    Attributes (EQUIVALENCEd onto BUNDLE_BLOCK)
      CHARACTER*(BUNDLE_LEN)    ATTRS
```

GpiSetBackColor - GSBCOL

```
CALL GSBCOL (HPS, COLOR)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Background color
      INTEGER*4   COLOR
```

GpiSetBackMix - GSBMIX

```
CALL GSBMIX (HPS, MIXMODE)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Background-mix mode
      INTEGER*4   MIXMODE
```

GpiSetBitmap - GSSBM

```
CALL GSSBM (HPS, HBM, OLD)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Handle of the bit map to be set
      INTEGER*4   HBM
C***                  Old bit-map handle
      INTEGER*4   OLD
```

GpiSetBitmapBits - GSBBIT

```
CALL GSBBIT (HPS, SCANSTART, SCANS, BUFFER, INFOTABLE, SCANSSET)

C***                                  Presentation-space handle
      INTEGER*4              HPS
C***                                  Line number
      INTEGER*4              SCANSTART
C***                                  Number of scan lines to be transmitted
      INTEGER*4              SCANS
C***                                  Bit-map data buffer
      CHARACTER*(n)          BUFFER
C***                                  Bit-map information table (EQUIVALENCEd onto
                                      BITINF_BLOCK)
      CHARACTER*(BITINF_LEN) INFOTABLE
C***                                  Number of scan lines actually set
      INTEGER*4              SCANSSET
```

GpiSetBitmapDimension - GSBDIM

```
CALL GSBDIM (HBM, WIDTH, HEIGHT)

C***                        Bit-map handle
      INTEGER*4   HBM
C***                        Width and height of bit map
      INTEGER*4   WIDTH, HEIGHT
```

GpiSetBitmapId - GSBMID

```
CALL GSBMID (HPS, HBM, LCID)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Bit-map handle
      INTEGER*4   HBM
C***                  Local identifier with which the bit map is to be tagged
      INTEGER*4   LCID
```

GpiSetCharAngle - GICA

```
      CALL GICA (HPS, X, Y)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Baseline angle
      INTEGER*4   X, Y
```

GpiSetCharBox - GICB

```
      CALL GICB (HPS, WIDTH, HEIGHT)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Character-box size in world coordinates
      INTEGER*4   WIDTH, HEIGHT
```

GpiSetCharDirection - GSCD

```
      CALL GSCD (HPS, DIRECTION)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Character direction
      INTEGER*4   DIRECTION
```

GpiSetCharMode - GSCM

```
      CALL GSCM (HPS, MODE)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Character mode
      INTEGER*4   MODE
```

GpiSetCharSet - GSCS

```
      CALL GSCS (HPS, LCID)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Character-set local identifier
      INTEGER*4   LCID
```

GpiSetCharShear - GICH

```
      CALL GICH (HPS, X, Y)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Character shear vector
      INTEGER*4   X, Y
```

GpiSetClipPath - GSSCLP

```
      CALL GSSCLP (HPS, PATH, OPTIONS)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Path control flag
      INTEGER*4   PATH
C***                    Options
      INTEGER*4   OPTIONS
```

GpiSetClipRegion - GSSCLR

```
      CALL GSSCLR (HPS, HRGN, OLD, COMPLEXITY)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Region handle
      INTEGER*4   HRGN
C***                          Old region handle (if any)
      INTEGER*4   OLD
C***                          Complexity of clipping/error indicator
      INTEGER*4   COMPLEXITY
```

GpiSetColor - GSCOL

```
      CALL GSCOL (HPS, COLOR)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Color
      INTEGER*4   COLOR
```

GpiSetCp - GSCPG

```
      CALL GSCPG (HPS, CODEPAGE)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Code-page id
      INTEGER*4   CODEPAGE
```

GpiSetCurrentPosition - GICP

```
      CALL GICP (HPS, X, Y)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  New value of current position
      INTEGER*4   X, Y
```

GpiSetDefArcParams - GIDAP

```
      CALL GIDAP (HPS, ARCPARAMS)

C***                                    Presentation-space handle
      INTEGER*4                HPS
C***                                    Default arc parameters (EQUIVALENCEd onto
                                        ARCPARML_BLOCK)
      CHARACTER*(ARCPARML_LEN)  ARCPARAMS
```

GpiSetDefAttrs - GIDATR

```
      CALL GIDATR (HPS, PRIMTYPE, ATTRMASK, ATTRS)

C***                                       Presentation-space handle
      INTEGER*4                 HPS
C***                                       Primitive type
      INTEGER*4                 PRIMTYPE
C***                                       Attributes mask
      INTEGER*4                 ATTRMASK
C***                                       Default attribute values (EQUIVALENCEd onto
                                           BUNDLE_BLOCK)
      CHARACTER*(BUNDLE_LEN)  ATTRS
```

GpiSetDefaultViewMatrix - GISDVM

```
CALL GISDVM (HPS, COUNT, ARRAY, OPTIONS)

C***                                        Presentation-space handle
      INTEGER*4                 HPS
C***                                        Number of elements
      INTEGER*4                 COUNT
C***                                        Transformation matrix (EQUIVALENCEd onto
                                            MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
C***                                        Transform options
      INTEGER*4                 OPTIONS
```

GpiSetDefTag - GSDTAG

```
CALL GSDTAG (HPS, TAG)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Default tag identifier
      INTEGER*4   TAG
```

GpiSetDefViewingLimits - GISDVL

```
CALL GISDVL (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Default viewing limits
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiSetDrawControl - GSDC

```
CALL GSDC (HPS, CONTROL, VALUE)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Drawing control
      INTEGER*4   CONTROL
C***                    Required value of the drawing control
      INTEGER*4   VALUE
```

GpiSetDrawingMode - GSDM

```
CALL GSDM (HPS, MODE)

C***                 Presentation-space handle
      INTEGER*4   HPS
C***                 Mode to be used for subsequent drawing calls
      INTEGER*4   MODE
```

GpiSetEditMode - GSEDM

```
CALL GSEDM (HPS, MODE)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Edit mode
      INTEGER*4   MODE
```

GpiSetElementPointer - GSEP

```
      CALL GSEP (HPS, ELEMENT)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        The element number required
      INTEGER*4   ELEMENT
```

GpiSetElementPointerAtLabel - GSEPLB

```
      CALL GSEPLB (HPS, LABEL)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Required label
      INTEGER*4   LABEL
```

GpiSetGraphicsField - GIFLD

```
      CALL GIFLD (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                         Presentation-space handle
      INTEGER*4   HPS
C***                        Graphics field
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiSetInitialSegmentAttrs - GSSATI

```
      CALL GSSATI (HPS, ATTRIBUTE, VALUE)

C***                           Presentation-space handle
      INTEGER*4   HPS
C***                           Segment attribute
      INTEGER*4   ATTRIBUTE
C***                           Attribute value
      INTEGER*4   VALUE
```

GpiSetLineEnd - GSLE

```
      CALL GSLE (HPS, LINEEND)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Style of line end
      INTEGER*4   LINEEND
```

GpiSetLineJoin - GSLJ

```
      CALL GSLJ (HPS, LINEJOIN)

C***                         Presentation-space handle
      INTEGER*4   HPS
C***                        Style of line join
      INTEGER*4   LINEJOIN
```

GpiSetLineType - GSLT

```
      CALL GSLT (HPS, LINETYPE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Line types available
      INTEGER*4   LINETYPE
```

GpiSetLineWidth - GILW

```
CALL GILW (HPS, LINEWIDTH)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Line-width multiplier
      INTEGER*4   LINEWIDTH
```

GpiSetLineWidthGeom - GIGLW

```
CALL GIGLW (HPS, LINEWIDTH)

C***                         Presentation-space handle
      INTEGER*4   HPS
C***                         Geometric line width
      INTEGER*4   LINEWIDTH
```

GpiSetMarker - GSMS

```
CALL GSMS (HPS, SYMBOL)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Marker symbol
      INTEGER*4   SYMBOL
```

GpiSetMarkerBox - GIMB

```
CALL GIMB (HPS, WIDTH, HEIGHT)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Size of marker box
      INTEGER*4   WIDTH, HEIGHT
```

GpiSetMarkerSet - GSMSS

```
CALL GSMSS (HPS, SET)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Marker-set local identifier
      INTEGER*4   SET
```

GpiSetMetaFileBits - GSMFB

```
CALL GSMFB (HMF, OFFSET, LENGTH, BUFFER)

C***                                Metafile-memory handle
      INTEGER*4           HMF
C***                                Offset
      INTEGER*4           OFFSET
C***                                Length of the metafile data
      INTEGER*4           LENGTH
C***                                Metafile data buffer
      CHARACTER*(length)  BUFFER
```

GpiSetMix - GSMIX

```
CALL GSMIX (HPS, MIXMODE)

C***                       Presentation-space handle
      INTEGER*4   HPS
C***                       Mix mode
      INTEGER*4   MIXMODE
```

GpiSetModelTransformMatrix - GISCTM

```
CALL GISCTM (HPS, COUNT, ARRAY, OPTIONS)

C***                                    Presentation-space handle
      INTEGER*4                HPS
C***                                    Number of elements in matrix
      INTEGER*4                COUNT
C***                                    Transformation matrix (EQUIVALENCEd onto
                                        MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN) ARRAY
C***                                    Transform options
      INTEGER*4                OPTIONS
```

GpiSetPageViewport - GIPV

```
CALL GIPV (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                  Presentation-space handle
      INTEGER*4  HPS
C***                  Page viewport
      INTEGER*4  XLEFT
      INTEGER*4  YBOTTOM
      INTEGER*4  XRIGHT
      INTEGER*4  YTOP
```

GpiSetPattern - GSPAT

```
CALL GSPAT (HPS, PATTERNSYMBOL)

C***                          Presentation-space handle
      INTEGER*4  HPS
C***                          Pattern symbol
      INTEGER*4  PATTERNSYMBOL
```

GpiSetPatternRefPoint - GIPRP

```
CALL GIPRP (HPS, X, Y)

C***                Presentation-space handle
      INTEGER*4  HPS
C***                Pattern reference point
      INTEGER*4  X, Y
```

GpiSetPatternSet - GSPATS

```
CALL GSPATS (HPS, SET)

C***                 Presentation-space handle
      INTEGER*4  HPS
C***                 Pattern-set local identifier
      INTEGER*4  SET
```

GpiSetPel - GIPEL

```
CALL GIPEL (HPS, X, Y)

C***                Presentation-space handle
      INTEGER*4  HPS
C***                Position in world coordinates
      INTEGER*4  X, Y
```

GpiSetPickAperturePosition - GIPAP

```
CALL GIPAP (HPS, X, Y)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Center of the pick aperture
      INTEGER*4   X, Y
```

GpiSetPickApertureSize - GISPAS

```
CALL GISPAS (HPS, OPTIONS, WIDTH, HEIGHT)

C***                            Presentation-space handle
      INTEGER*4   HPS
C***                            Setting option
      INTEGER*4   OPTIONS
C***                            Pick aperture size
      INTEGER*4   WIDTH, HEIGHT
```

GpiSetPS - GIPS

```
CALL GIPS (HPS, WIDTH, HEIGHT, OPTIONS)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Presentation-space size
      INTEGER*4   WIDTH, HEIGHT
C***                          Options
      INTEGER*4   OPTIONS
```

GpiSetRegion - GIREGN

```
CALL GIREGN (HPS, HRGN, COUNT, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                              Presentation-space handle
      INTEGER*4   HPS
C***                              Region handle
      INTEGER*4   HRGN
C***                              Count of rectangles
      INTEGER*4   COUNT
C***                              Array of rectangles
      INTEGER*4   XLEFT(count)
      INTEGER*4   YBOTTOM(count)
      INTEGER*4   XRIGHT(count)
      INTEGER*4   YTOP(count)
```

GpiSetSegmentAttrs - GSSATS

```
CALL GSSATS (HPS, SEGID, ATTRIBUTE, VALUE)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Segment identifier
      INTEGER*4   SEGID
C***                        Segment attribute
      INTEGER*4   ATTRIBUTE
C***                        Attribute value
      INTEGER*4   VALUE
```

GpiSetSegmentPriority - GSSPRI

```
      CALL GSSPRI (HPS, SEGID, REFSEGID, ORDER)

C***                        Presentation-space handle
      INTEGER*4   HPS
C***                        Segment identifier
      INTEGER*4   SEGID
C***                        Reference segment identifier
      INTEGER*4   REFSEGID
C***                        Segment higher or lower
      INTEGER*4   ORDER
```

GpiSetSegmentTransformMatrix - GISTFM

```
      CALL GISTFM (HPS, SEGID, COUNT, ARRAY, OPTIONS)

C***                                        Presentation-space handle
      INTEGER*4                 HPS
C***                                        Segment identifier
      INTEGER*4                 SEGID
C***                                        Number of elements
      INTEGER*4                 COUNT
C***                                        Transformation matrix (EQUIVALENCEd onto
                                            MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)  ARRAY
C***                                        Transform options
      INTEGER*4                 OPTIONS
```

GpiSetStopDraw - GSSDW

```
      CALL GSSDW (HPS, VALUE)

C***                  Presentation-space handle
      INTEGER*4   HPS
C***                  Stop draw condition
      INTEGER*4   VALUE
```

GpiSetTag - GSTAG

```
      CALL GSTAG (HPS, TAG)

C***                Presentation-space handle
      INTEGER*4   HPS
C***                Tag identifier
      INTEGER*4   TAG
```

GpiSetViewingLimits - GISVL

```
      CALL GISVL (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP)

C***                    Presentation-space handle
      INTEGER*4   HPS
C***                    Viewing limits in model space
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
```

GpiSetViewingTransformMatrix - GISVTM

```
CALL GISVTM (HPS, COUNT, ARRAY, OPTIONS)

C***                                        Presentation-space handle
      INTEGER*4                  HPS
C***                                        Number of elements
      INTEGER*4                  COUNT
C***                                        Transformation matrix (EQUIVALENCEd onto
                                            MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)   ARRAY
C***                                        Transform option
      INTEGER*4                  OPTIONS
```

GpiStrokePath - GSSTRP

```
CALL GSSTRP (HPS, PATH, OPTIONS)

C***                      Presentation-space handle
      INTEGER*4   HPS
C***                      Identifier of path to be stroked; it must be 1
      INTEGER*4   PATH
C***                      Stroke option
      INTEGER*4   OPTIONS
```

GpiTranslate - GIXLAT

```
CALL GIXLAT (HPS, ARRAY, OPTIONS, X, Y)

C***                                        Presentation-space handle
      INTEGER*4                  HPS
C***                                        Transform matrix (EQUIVALENCEd onto MATRIXLF_BLOCK)
      CHARACTER*(MATRIXLF_LEN)   ARRAY
C***                                        Transform options
      INTEGER*4                  OPTIONS
C***                                        Translation
      INTEGER*4                  X, Y
```

GpiUnloadFonts - GSFUL

```
CALL GSFUL (HAB, LENGTH, STRING)

C***                            Anchor-block handle
      INTEGER*4           HAB
C***                            Fully qualified file name of the font resource
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
```

GpiUnrealizeColorTable - GSUCT

```
CALL GSUCT (HPS)

C***                 Presentation-space handle
      INTEGER*4   HPS
```

GpiWCBitBlt - GIWBB

```
      CALL GIWBB (TARGET, SOURCE, COUNT, X, Y, ROP, OPTIONS)

C***                                  Target presentation-space handle
      INTEGER*4   TARGET
C***                                  Source bit-map handle
      INTEGER*4   SOURCE
C***                                  Point count
      INTEGER*4   COUNT
C***                                  Point array
      INTEGER*4   X(count), Y(count)
C***                                  Mixing function required
      INTEGER*4   ROP
C***                                  Options
      INTEGER*4   OPTIONS
```

Chapter 5. Picture Function Calls

PicIchg - PICONV

```
CALL PICONV (HAB, LENGTH, STRING, LENGTH_2, STRING_2, TYPE, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    The name of the source file
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    The name of the destination file
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Specifies the type of conversion
      INTEGER*4             TYPE
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

PicPrint - PIPRT

```
CALL PIPRT (HAB, LENGTH, STRING, TYPE, LENGTH_2, STRING_2, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Full path and filename of the source file
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Type of source file
      INTEGER*4             TYPE
C***                                    Spooler parameters
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

Chapter 6. Profile Function Calls

PrfAddProgram - PRADPR

```
CALL PRADPR (HINI, DETAILS, GROUP, PROGRAM)

C***                    Initialization-file handle
     INTEGER*4   HINI
C***                    Program details for the program to be added to the program list
                        (PROGDETAILS→HSTRUCT)
     INTEGER*4   DETAILS
C***                    Handle of the group to which the program is to be added
     INTEGER*4   GROUP
C***                    Program handle for the program added to the program list
     INTEGER*4   PROGRAM
```

PrfChangeProgram - PRCPRG

```
CALL PRCPRG (HINI, PROGRAM, DETAILS, SUCCESS)

C***                    Initialization-file handle
     INTEGER*4   HINI
C***                    Handle of the program whose information is to be changed
     INTEGER*4   PROGRAM
C***                    Program details (PROGDETAILS→HSTRUCT)
     INTEGER*4   DETAILS
C***                    Success indicator
     INTEGER*4   SUCCESS
```

PrfCloseProfile - PRCPR

```
CALL PRCPR (HINI, SUCCESS)

C***                   Initialization-file handle
     INTEGER*4   HINI
C***                   Success indicator
     INTEGER*4   SUCCESS
```

PrfCreateGroup - PRCGRP

```
CALL PRCGRP (HINI, LENGTH, STRING, VISIBILITY, GROUP)

C***                                Initialization-file handle
     INTEGER*4           HINI
C***                                Title of the new group
     INTEGER*4           LENGTH
     CHARACTER*(length)  STRING
C***                                Visibility control
     INTEGER*4           VISIBILITY
C***                                Group handle for the newly-created group
     INTEGER*4           GROUP
```

PrfDestroyGroup - PRDGRP

```
CALL PRDGRP (HINI, GROUP, SUCCESS)

C***                    Initialization-file handle
     INTEGER*4   HINI
C***                    Group handle
     INTEGER*4   GROUP
C***                    Success indicator
     INTEGER*4   SUCCESS
```

PrfOpenProfile - PROPPR

```
      CALL PROPPR (HAB, LENGTH, STRING, HINI)

C***                                  Anchor-block handle
      INTEGER*4           HAB
C***                                  User-profile file name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                  Initialization-file handle
      INTEGER*4           HINI
```

PrfQueryDefinition - PRQDEF

```
      CALL PRQDEF (HINI, PROGRAM, DETAILS, MAXLEN, RETLEN)

C***                      Initialization-file handle
      INTEGER*4   HINI
C***                      Handle of the program whose details are to be returned
      INTEGER*4   PROGRAM
C***                      Program details (PROGDETAILS→HSTRUCT)
      INTEGER*4   DETAILS
C***                      Maximum length in bytes
      INTEGER*4   MAXLEN
C***                      Length of returned data
      INTEGER*4   RETLEN
```

PrfQueryProfile - PRQPR

```
      CALL PRQPR (HAB, PROFILE, SUCCESS)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Profile names structure (PRFPROFILE→HSTRUCT)
      INTEGER*4   PROFILE
C***                    Success indicator
      INTEGER*4   SUCCESS
```

PrfQueryProfileData - PRQPDA

```
      CALL PRQPDA (HINI, LENGTH, STRING, LENGTH_2, STRING_2, VALUE, SIZE, SUCCESS)

C***                                  Initialization-file handle
      INTEGER*4             HINI
C***                                  Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                  Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                  Value data (STORAGE→HSTRUCT)
      INTEGER*4             VALUE
C***                                  Size of value data
      INTEGER*4             SIZE
C***                                  Success indicator
      INTEGER*4             SUCCESS
```

PrfQueryProfileInt - PRQPIN

```
CALL PRQPIN (HINI, LENGTH, STRING, LENGTH_2, STRING_2, DEFAULT, RESULT)

C***                                    Initialization-file handle
      INTEGER*4             HINI
C***                                    Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Default value
      INTEGER*4             DEFAULT
C***                                    Key value specified in the intialization file
      INTEGER*4             RESULT
```

PrfQueryProfileSize - PRQPSZ

```
CALL PRQPSZ (HINI, LENGTH, STRING, LENGTH_2, STRING_2, DATALEN, SUCCESS)

C***                                    Initialization-file handle
      INTEGER*4             HINI
C***                                    Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Data length
      INTEGER*4             DATALEN
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

PrfQueryProfileString - PRQPST

```
CALL PRQPST (HINI, LENGTH, STRING, LENGTH_2, STRING_2, LENGTH_3, STRING_3,
             PROFILESTRING, MAXLEN, RETLEN)

C***                                    Initialization-file handle
      INTEGER*4             HINI
C***                                    Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Default string
      INTEGER*4             LENGTH_3
      CHARACTER*(length_3)  STRING_3
C***                                    Profile string (STORAGE→HSTRUCT)
      INTEGER*4             PROFILESTRING
C***                                    Maximum string length
      INTEGER*4             MAXLEN
C***                                    String length returned
      INTEGER*4             RETLEN
```

PrfQueryProgramCategory - PRQPC

```
      CALL PRQPC (HINI, LENGTH, STRING, CATEGORY)

C***                                  Initialization-file handle
      INTEGER*4           HINI
C***                                  Executable-file name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                  Program category
      INTEGER*4           CATEGORY
```

PrfQueryProgramHandle - PRQPHD

```
      CALL PRQPHD (HINI, LENGTH, STRING, PROGARRAY, MAXLEN, COUNT, RETLEN)

C***                                      Initialization-file handle
      INTEGER*4               HINI
C***                                      Executable-file name
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                      Program handles (EQUIVALENCEd onto HPROGA_BLOCK)
      CHARACTER*(HPROGA_LEN)  PROGARRAY
C***                                      Maximum number of program handles
      INTEGER*4               MAXLEN
C***                                      Number of program handles returned
      INTEGER*4               COUNT
C***                                      Number of bytes returned
      INTEGER*4               RETLEN
```

PrfQueryProgramTitles - PRQPTI

```
      CALL PRQPTI (HINI, GROUP, TITLES, MAXLEN, COUNT, RETLEN)

C***                              Initialization-file handle
      INTEGER*4   HINI
C***                              Handle of the program or group whose information is to be
                                  returned
      INTEGER*4   GROUP
C***                              Program information buffer (PROGTITLE→HSTRUCT)
      INTEGER*4   TITLES(count)
C***                              Buffer length
      INTEGER*4   MAXLEN
C***                              Number of structures returned
      INTEGER*4   COUNT
C***                              Length returned
      INTEGER*4   RETLEN
```

PrfRemoveProgram - PRRPRG

```
      CALL PRRPRG (HINI, PROGRAM, SUCCESS)

C***                        Initialization-file handle
      INTEGER*4   HINI
C***                        Program handle
      INTEGER*4   PROGRAM
C***                        Success indicator
      INTEGER*4   SUCCESS
```

PrfReset - PRRES

```
CALL PRRES (HAB, PROFILE, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Profile-names structure (PRFPROFILE→HSTRUCT)
      INTEGER*4   PROFILE
C***                        Success indicator
      INTEGER*4   SUCCESS
```

PrfWriteProfileData - PRWPDA

```
CALL PRWPDA (HINI, LENGTH, STRING, LENGTH_2, STRING_2, VALUE, SIZE, SUCCESS)

C***                                  Initialization-file handle
      INTEGER*4              HINI
C***                                  Application name
      INTEGER*4              LENGTH
      CHARACTER*(length)     STRING
C***                                  Key name
      INTEGER*4              LENGTH_2
      CHARACTER*(length_2)   STRING_2
C***                                  Value data (STORAGE→HSTRUCT)
      INTEGER*4              VALUE
C***                                  Size of value data
      INTEGER*4              SIZE
C***                                  Success indicator
      INTEGER*4              SUCCESS
```

PrfWriteProfileString - PRWPST

```
CALL PRWPST (HINI, LENGTH, STRING, LENGTH_2, STRING_2, LENGTH_3, STRING_3, SUCCESS)

C***                                  Initialization-file handle
      INTEGER*4              HINI
C***                                  Application name
      INTEGER*4              LENGTH
      CHARACTER*(length)     STRING
C***                                  Key name
      INTEGER*4              LENGTH_2
      CHARACTER*(length_2)   STRING_2
C***                                  Text string
      INTEGER*4              LENGTH_3
      CHARACTER*(length_3)   STRING_3
C***                                  Success indicator
      INTEGER*4              SUCCESS
```

Chapter 7. Spooler Function Calls

SplQmAbort - SMABT

```
      CALL SMABT (HSPL, SUCCESS)

C***                          Spooler handle
      INTEGER*4   HSPL
C***                          Success indicator
      INTEGER*4   SUCCESS
```

SplQmClose - SMCLS

```
      CALL SMCLS (HSPL, SUCCESS)

C***                          Spooler handle
      INTEGER*4   HSPL
C***                          Success indicator
      INTEGER*4   SUCCESS
```

SplQmEndDoc - SMEDOC

```
      CALL SMEDOC (HSPL, JOB)

C***                         Spooler handle
      INTEGER*4   HSPL
C***                         Job identifier
      INTEGER*4   JOB
```

SplQmOpen - SMOPEN

```
      CALL SMOPEN (LENGTH, STRING, COUNT, DATA, HSPL)

C***                                      A token (nickname) that identifies spooler
                                          information
      INTEGER*4                LENGTH
      CHARACTER*(length)       STRING
C***                                      Number of items
      INTEGER*4                COUNT
C***                                      Open parameters (EQUIVALENCEd onto DEVSTRUC_BLOCK)
      CHARACTER*(DEVSTRUC_LEN) DATA
C***                                      Spooler handle
      INTEGER*4                HSPL
```

SplQmStartDoc - SMSDOC

```
      CALL SMSDOC (HSPL, LENGTH, STRING, SUCCESS)

C***                              Spooler handle
      INTEGER*4          HSPL
C***                              Document name
      INTEGER*4          LENGTH
      CHARACTER*(length) STRING
C***                              Success indicator
      INTEGER*4          SUCCESS
```

SplQmWrite - SMWRT

```
      CALL SMWRT (HSPL, COUNT, DATA, SUCCESS)

C***                                  Spooler handle
      INTEGER*4          HSPL
C***                                  Length in bytes
      INTEGER*4          COUNT
C***                                  Buffer of data to be written to the spool file
      CHARACTER*(count)  DATA
C***                                  Success indicator
      INTEGER*4          SUCCESS
```

SplQpInstall - SPINST

```
      CALL SPINST (HWND, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Success indicator
      INTEGER*4   SUCCESS
```

SplQpQueryDt - SPQDT

```
      CALL SPQDT (COUNT, DATATYPES, SUCCESS)

C***                              Maximum number of data types that can be returned
      INTEGER*4   COUNT
C***                              An array containing the data types supported
                                  (STRL→HSTRUCT)
      INTEGER*4   DATATYPES(count)
C***                              Success indicator
      INTEGER*4   SUCCESS
```

Chapter 8. Window Function Calls

WinAddAtom - WUADAT

```
CALL WUADAT (ATOMTBL, LENGTH, STRING, ATOM)

C***                                  Atom-table handle
      INTEGER*4           ATOMTBL
C***                                  Atom name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                  Atom value
      INTEGER*4           ATOM
```

WinAddProgram - WMADPR

```
CALL WMADPR (HAB, PROGRAMINFO, GROUPHANDLE, PROGHANDLE)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Program information for the program to be added to the
                            installed program list (PIBSTRUCT->HSTRUCT)
      INTEGER*4   PROGRAMINFO
C***                        Handle of the program group to which the program is to be
                            added
      INTEGER*4   GROUPHANDLE
C***                        Program handle for the program added to the installed program
                            list
      INTEGER*4   PROGHANDLE
```

WinAddSwitchEntry - WMASWE

```
CALL WMASWE (SWITCHDATA, SWITCH)

C***                                      Switch data (EQUIVALENCEd onto SWCNTRL_BLOCK)
      CHARACTER*(SWCNTRL_LEN)   SWITCHDATA
C***                                      Handle to the newly-created switch list entry
      INTEGER*4                 SWITCH
```

WinAlarm - WMALRM

```
CALL WMALRM (DESKTOP, STYLE, RESULT)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Alarm style
      INTEGER*4   STYLE
C***                        Alarm-generated indicator
      INTEGER*4   RESULT
```

WinBeginEnumWindows - WIBEWN

```
CALL WIBEWN (PARENT, HENUM)

C***                        Handle of the window whose child windows are to be enumerated
      INTEGER*4   PARENT
C***                        Enumeration handle
      INTEGER*4   HENUM
```

WinBeginPaint - WIBPNT

```
      CALL WIBPNT (HWND, HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, PAINTPS)

C***                          Handle of window where drawing is going to occur
      INTEGER*4   HWND
C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Bounding rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Presentation-space handle
      INTEGER*4   PAINTPS
```

WinBroadcastMsg - WMBRMG

```
      CALL WMBRMG (PARENT, MSGID, PARAM1, PARAM2, CMD, SUCCESS)

C***                          Parent-window handle
      INTEGER*4   PARENT
C***                          Message identifier
      INTEGER*4   MSGID
C***                          Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                          Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                          Broadcast message command
      INTEGER*4   CMD
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinCalcFrameRect - WICRCT

```
      CALL WICRCT (HWND, XLEFT, YBOTTOM, XRIGHT, YTOP, FRAME, SUCCESS)

C***                          Frame-window handle
      INTEGER*4   HWND
C***                          Window rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Frame indicator
      INTEGER*4   FRAME
C***                          Rectangle-calculated indicator
      INTEGER*4   SUCCESS
```

WinCallMsgFilter - WMCMFI

```
      CALL WMCMFI (HAB, PQMSG, FILTER, HOOKRET)

C***                                Anchor-block handle
      INTEGER*4             HAB
C***                                Message to be passed to the message-filter hook
                                    (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)  PQMSG
C***                                Filter
      INTEGER*4             FILTER
C***                                Message-filter hook return indicator
      INTEGER*4             HOOKRET
```

WinCancelShutdown - WMCSD

```
CALL WMCSD (HMQ, CANCELALWAYS, SUCCESS)

C***                          Queue handle
      INTEGER*4   HMQ
C***                          Cancellation control
      INTEGER*4   CANCELALWAYS
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinChangeSwitchEntry - WMCSWE

```
CALL WMCSWE (SWITCH, SWITCHDATA, RETCODE)

C***                                    Handle to the switch-list entry to be changed
      INTEGER*4                SWITCH
C***                                    Switch-control data (EQUIVALENCEd onto
                                        SWCNTRL_BLOCK)
      CHARACTER*(SWCNTRL_LEN)  SWITCHDATA
C***                                    Return code
      INTEGER*4                RETCODE
```

WinCloseClipbrd - WICCBD

```
CALL WICCBD (HAB, SUCCESS)

C***                     Anchor-block handle
      INTEGER*4   HAB
C***                     Success indicator
      INTEGER*4   SUCCESS
```

WinCompareStrings - WUCOMP

```
CALL WUCOMP (HAB, CODEPAGE, COUNTRYCODE, LENGTH, STRING, LENGTH_2, STRING_2, OPTIONS,
             RESULT)

C***                                  Anchor-block handle
      INTEGER*4            HAB
C***                                  Codepage identity of both strings
      INTEGER*4            CODEPAGE
C***                                  Country code
      INTEGER*4            COUNTRYCODE
C***                                  String 1
      INTEGER*4            LENGTH
      CHARACTER*(length)   STRING
C***                                  String 2
      INTEGER*4            LENGTH_2
      CHARACTER*(length_2) STRING_2
C***                                  Reserved
      INTEGER*4            OPTIONS
C***                                  Comparison result
      INTEGER*4            RESULT
```

WinCopyAccelTable - WMCAT

```
CALL WMCAT (HACCEL, ACCELTABLE, COPYMAX, COPIED)

C***                                    Accelerator-table handle
      INTEGER*4                HACCEL
C***                                    Accelerator-table data area (EQUIVALENCEd onto
                                        ACCELTAB_BLOCK)
      CHARACTER*(ACCELTAB_LEN) ACCELTABLE
C***                                    Maximum data area size
      INTEGER*4                COPYMAX
C***                                    Amount copied or size required
      INTEGER*4                COPIED
```

WinCopyRect - WICR

```
      CALL WICR (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, XLEFT_2, YBOTTOM_2, XRIGHT_2, YTOP_2,
                 SUCCESS)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Destination rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Source rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinCpTranslateChar - WUTRCH

```
      CALL WUTRCH (HAB, CPSOURCE, SOURCE, CPDEST, DEST)

C***                           Anchor-block handle
      INTEGER*4   HAB
C***                           Source-character code page
      INTEGER*4   CPSOURCE
C***                           Character to be translated
      INTEGER*4   SOURCE
C***                           Code page of the resultant character
      INTEGER*4   CPDEST
C***                           If nonzero, the translated character
      INTEGER*4   DEST
```

WinCpTranslateString - WUTRST

```
      CALL WUTRST (HAB, CPSOURCE, LENGTH, STRING, CPDEST, LENDEST, LENGTH_2, STRING_2,
                   SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4            HAB
C***                                    Source-string code page
      INTEGER*4            CPSOURCE
C***                                    String to be translated
      INTEGER*4            LENGTH
      CHARACTER*(length)   STRING
C***                                    Code page of the resultant string
      INTEGER*4            CPDEST
C***                                    Maximum length of output string
      INTEGER*4            LENDEST
C***                                    The translated string (LENGTH_2 is result string length)
      INTEGER*4            LENGTH_2
      CHARACTER*(lendest)  STRING_2
C***                                    Success indicator
      INTEGER*4            SUCCESS
```

WinCreateAccelTable - WMCACT

```
      CALL WMCACT (HAB, ACCELTABLE, HACCEL)

C***                                          Anchor-block handle
      INTEGER*4                 HAB
C***                                          Accelerator table (EQUIVALENCEd onto
                                              ACCELTAB_BLOCK)
      CHARACTER*(ACCELTAB_LEN)  ACCELTABLE
C***                                          Accelerator-table handle
      INTEGER*4                 HACCEL
```

WinCreateAtomTable - WMCATT

```
      CALL WMCATT (INITIAL, BUCKETS, ATOMTBL)

C***                        Initial bytes
      INTEGER*4   INITIAL
C***                        Size of the hash table
      INTEGER*4   BUCKETS
C***                        Atom-table handle
      INTEGER*4   ATOMTBL
```

WinCreateCursor - WMCCUR

```
      CALL WMCCUR (HWND, X, Y, CX, CY, RGF, XLEFT, YBOTTOM, XRIGHT, YTOP, SUCCESS)

C***                       Handle of window in which cursor is displayed
      INTEGER*4   HWND
C***                       x position of cursor
      INTEGER*4   X
C***                       y position of cursor
      INTEGER*4   Y
C***                       x size of cursor
      INTEGER*4   CX
C***                       y size of cursor
      INTEGER*4   CY
C***                       Controls the appearance of the cursor
      INTEGER*4   RGF
C***                       Cursor rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                       Success indicator
      INTEGER*4   SUCCESS
```

WinCreateDataStructure - WUCDS

```
CALL WUCDS (HAB, COUNT, TYPES, INLEN, INSTRUC, MAXLEN, HANDLE, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4           HAB
C***                                    Number of elements
      INTEGER*4           COUNT
C***                                    Data type codes
      INTEGER*4           TYPES(count)
C***                                    Length of input structure
      INTEGER*4           INLEN
C***                                    Input structure
      CHARACTER*(inlen)   INSTRUC
C***                                    Maximum length of structure
      INTEGER*4           MAXLEN
C***                                    Handle of data structure created
      INTEGER*4           HANDLE
C***                                    Success indicator
      INTEGER*4           SUCCESS
```

WinCreateDlg - WCCRD

```
CALL WCCRD (PARENT, OWNER, DLGPROC, DLGTMP, CREATEPARAMS, DLG)

C***                                    Parent-window handle of the created dialog window
      INTEGER*4             PARENT
C***                                    Requested owner-window handle of the created dialog
                                        window
      INTEGER*4             OWNER
C***                                    Dialog procedure for the created dialog window
      INTEGER*4             DLGPROC
C***                                    Dialog template (EQUIVALENCEd onto DLGT_BLOCK)
      CHARACTER*(DLGT_LEN)  DLGTMP
C***                                    Application-defined data area (CREATEPARAMS->HSTRUCT)
      INTEGER*4             CREATEPARAMS
C***                                    Dialog-window handle
      INTEGER*4             DLG
```

WinCreateFrameControls - WIFRCO

```
CALL WIFRCO (FRAME, FCDATA, LENGTH, STRING, SUCCESS)

C***                                    Frame-window handle
      INTEGER*4               FRAME
C***                                    Frame-control data (EQUIVALENCEd onto
                                        FRACDATA_BLOCK)
      CHARACTER*(FRACDATA_LEN)  FCDATA
C***                                    Title string
      INTEGER*4               LENGTH
      CHARACTER*(length)      STRING
C***                                    Success indicator
      INTEGER*4               SUCCESS
```

WinCreateGroup - WMCGRP

```
CALL WMCGRP (HAB, LENGTH, STRING, VISIBILITY, RES1, RES2, GROUPHANDLE)

C***                                    Anchor-block handle
      INTEGER*4           HAB
C***                                    Title of the new group
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                    Visibility control
      INTEGER*4           VISIBILITY
C***                                    Reserved
      INTEGER*4           RES1
C***                                    Reserved
      INTEGER*4           RES2
C***                                    Program-group handle for the newly-created group
      INTEGER*4           GROUPHANDLE
```

WinCreateMsgQueue - WMCMQ

```
CALL WMCMQ (HAB, QUEUESIZE, HMQ)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Maximum queue size
      INTEGER*4   QUEUESIZE
C***                        Message-queue handle
      INTEGER*4   HMQ
```

WinCreatePointer - WICPOI

```
CALL WICPOI (DESKTOP, BITMAP, POINTERSIZE, XHOTSPOT, YHOTSPOT, HPTR)

C***                            Desktop-window handle or HWND_DESKTOP
      INTEGER*4   DESKTOP
C***                            Bit-map handle from which the pointer image is created
      INTEGER*4   BITMAP
C***                            Pointer-size indicator
      INTEGER*4   POINTERSIZE
C***                            x offset of hotspot within pointer from its lower left corner
                                (in pels)
      INTEGER*4   XHOTSPOT
C***                            y offset of hotspot within pointer from its lower left corner
                                (in pels)
      INTEGER*4   YHOTSPOT
C***                            Pointer handle
      INTEGER*4   HPTR
```

WinCreatePointerIndirect - WICPIN

```
CALL WICPIN (DESKTOP, POINTERINFO, HPTR)

C***                                    Desktop-window handle or HWND_DESKTOP
      INTEGER*4               DESKTOP
C***                                    Pointer information structure (EQUIVALENCEd onto
                                        PTRINF_BLOCK)
      CHARACTER*(PTRINF_LEN)  POINTERINFO
C***                                    Pointer handle
      INTEGER*4               HPTR
```

WinCreateStdWindow - WICRTS

```
      CALL WICRTS (PARENT, STYLE, CREATEFLAGS, LENGTH, STRING, LENGTH_2, STRING_2,
                   STYLECLIENT, RESOURCE, ID, CLIENT, FRAME)

C***                                        Parent-window handle
      INTEGER*4              PARENT
C***                                        Frame-window style
      INTEGER*4              STYLE
C***                                        Frame-creation flags
      INTEGER*4              CREATEFLAGS
C***                                        Client-window class name
      INTEGER*4              LENGTH
      CHARACTER*(length)     STRING
C***                                        Title-bar text
      INTEGER*4              LENGTH_2
      CHARACTER*(length_2)   STRING_2
C***                                        Client-window style
      INTEGER*4              STYLECLIENT
C***                                        Resource identifier
      INTEGER*4              RESOURCE
C***                                        Frame-window identifier
      INTEGER*4              ID
C***                                        Client-window handle
      INTEGER*4              CLIENT
C***                                        Frame-window handle
      INTEGER*4              FRAME
```

WinCreateSwitchEntry - WMCRSE

```
      CALL WMCRSE (HAB, SWITCHDATA, SWITCH)

C***                                        Anchor-block handle
      INTEGER*4                HAB
C***                                        Switch data (EQUIVALENCEd onto SWCNTRL_BLOCK)
      CHARACTER*(SWCNTRL_LEN)  SWITCHDATA
C***                                        Handle to the newly-created switch list entry
      INTEGER*4                SWITCH
```

WinCreateWindow - WICRT

```
CALL WICRT (PARENT, LENGTH, STRING, LENGTH_2, STRING_2, STYLE, XCOORD, YCOORD, WIDTH,
            HEIGHT, OWNER, BEHIND, ID, CTLDATA, PRESPARAMS, HWND)

C***                                    Parent-window handle
      INTEGER*4             PARENT
C***                                    Registered-class name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Window text
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Window style
      INTEGER*4             STYLE
C***                                    x coordinate of window position
      INTEGER*4             XCOORD
C***                                    y coordinate of window position
      INTEGER*4             YCOORD
C***                                    Width of window, in window coordinates
      INTEGER*4             WIDTH
C***                                    Height of window, in window coordinates
      INTEGER*4             HEIGHT
C***                                    Owner-window handle
      INTEGER*4             OWNER
C***                                    Sibling-window handle
      INTEGER*4             BEHIND
C***                                    Window identifier
      INTEGER*4             ID
C***                                    Control data (CTLDATA->HSTRUCT)
      INTEGER*4             CTLDATA
C***                                    Presentation parameters (PRESDATA->HSTRUCT)
      INTEGER*4             PRESPARAMS
C***                                    Window handle
      INTEGER*4             HWND
```

WinDefDlgProc - WCDDP

```
CALL WCDDP (DLG, MSGID, PARAM1, PARAM2, REPLY)

C***                  Dialog-window handle
      INTEGER*4   DLG
C***                  Message identity
      INTEGER*4   MSGID
C***                  Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                  Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                  Message-return data (MRESULT: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   REPLY
```

WinDefWindowProc - WMDWP

```
CALL WMDWP (HWND, MSGID, PARAM1, PARAM2, REPLY)

C***                  Window handle
      INTEGER*4   HWND
C***                  Message identity
      INTEGER*4   MSGID
C***                  Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                  Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                  Message-return data (MRESULT: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   REPLY
```

WinDeleteAtom - WMDELA

```
CALL WMDELA (ATOMTBL, ATOM, RETURNCODE)

C***                          Atom-table handle
      INTEGER*4   ATOMTBL
C***                          Atom identifying the atom to be deleted
      INTEGER*4   ATOM
C***                          Return code
      INTEGER*4   RETURNCODE
```

WinDeleteLibrary - WMDLIB

```
CALL WMDLIB (HAB, LIBHANDLE, DELETED)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Library handle to be deleted
      INTEGER*4   LIBHANDLE
C***                          Library-deleted indicator
      INTEGER*4   DELETED
```

WinDeleteProcedure - WMDPR

```
CALL WMDPR (HAB, WNDPROC, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window procedure identifier to be deleted
      INTEGER*4   WNDPROC
C***                        Procedure-deleted indicator
      INTEGER*4   SUCCESS
```

WinDestroyAccelTable - WMDACT

```
CALL WMDACT (ACCEL, SUCCESS)

C***                         Accelerator-table handle
      INTEGER*4   ACCEL
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinDestroyAtomTable - WMDATT

```
CALL WMDATT (ATOMTBL, RETURNCODE)

C***                          Atom-table handle
      INTEGER*4   ATOMTBL
C***                          Return code
      INTEGER*4   RETURNCODE
```

WinDestroyCursor - WMDCUR

```
CALL WMDCUR (HWND, SUCCESS)

C***                        Window handle to which the cursor belongs
      INTEGER*4   HWND
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinDestroyDataStructure - WUDDS

```
CALL WUDDS (HAB, HANDLE, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Data structure handle
      INTEGER*4   HANDLE
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinDestroyMsgQueue - WMDMQ

```
CALL WMDMQ (HMQ, DESTROYED)

C***                          Message-queue handle
      INTEGER*4   HMQ
C***                          Queue-destroyed indicator
      INTEGER*4   DESTROYED
```

WinDestroyPointer - WMDPOI

```
CALL WMDPOI (POINTER, SUCCESS)

C***                          Handle of pointer to be destroyed
      INTEGER*4   POINTER
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinDestroyWindow - WIDEL

```
CALL WIDEL (HWND, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Window-destroyed indicator
      INTEGER*4   SUCCESS
```

WinDismissDlg - WCDELD

```
CALL WCDELD (DLG, RESULT, SUCCESS)

C***                          Dialog-window handle
      INTEGER*4   DLG
C***                          Reply value
      INTEGER*4   RESULT
C***                          Dialog-dismissed indicator
      INTEGER*4   SUCCESS
```

WinDispatchMsg - WMDISP

```
CALL WMDISP (HAB, MSG, REPLY)

C***                                Anchor-block handle
      INTEGER*4             HAB
C***                                Message structure (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)  MSG
C***                                Message-return data (MRESULT: Can also be HSTRUCT or
                                    COMPOSED)
      INTEGER*4             REPLY
```

WinDlgBox - WCLMD

```
      CALL WCLMD (PARENT, OWNER, DLGPROC, RESOURCE, DLGID, CREATEPARAMS, RESULT)

C***                                Parent-window handle of the created dialog window
      INTEGER*4   PARENT
C***                                Requested owner-window handle of the created dialog window
      INTEGER*4   OWNER
C***                                Dialog procedure for the created dialog window
      INTEGER*4   DLGPROC
C***                                Resource identity containing the dialog template
      INTEGER*4   RESOURCE
C***                                Dialog-template identity within the resource file
      INTEGER*4   DLGID
C***                                Application-defined data area (CREATEPARAMS→HSTRUCT)
      INTEGER*4   CREATEPARAMS
C***                                Reply value
      INTEGER*4   RESULT
```

WinDrawBitmap - WIDBMP

```
      CALL WIDBMP (HPS, HBM, XLEFT, YBOTTOM, XRIGHT, YTOP, XLEFT_2, YBOTTOM_2, XRIGHT_2,
                   YTOP_2, FORECOLOR, BACKCOLOR, RGF, SUCCESS)

C***                                Handle of presentation space in which the bit map is drawn
      INTEGER*4   HPS
C***                                Bit-map handle
      INTEGER*4   HBM
C***                                Subrectangle of bit map to be drawn
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                                Bit-map destination
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                                Foreground color
      INTEGER*4   FORECOLOR
C***                                Background color
      INTEGER*4   BACKCOLOR
C***                                Flags that determine how the bit map is drawn
      INTEGER*4   RGF
C***                                Success indicator
      INTEGER*4   SUCCESS
```

WinDrawBorder - WIBORD

```
      CALL WIBORD (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, VERTSIDEWIDTH, HORIZSIDEWIDTH,
                   BORDERCOLOR, INTERIORCOLOR, CMD, SUCCESS)

C***                                  Presentation-space handle
      INTEGER*4   HPS
C***                                  Bounding rectangle for the border
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                                  Width of border rectangle vertical sides
      INTEGER*4   VERTSIDEWIDTH
C***                                  Width of border rectangle horizontal sides
      INTEGER*4   HORIZSIDEWIDTH
C***                                  Color of edge of border
      INTEGER*4   BORDERCOLOR
C***                                  Color of interior of border
      INTEGER*4   INTERIORCOLOR
C***                                  Flags controlling the way in which the border is drawn
      INTEGER*4   CMD
C***                                  Success indicator
      INTEGER*4   SUCCESS
```

WinDrawPointer - WMDRPT

```
      CALL WMDRPT (HPS, X, Y, POINTER, HALFTONE, SUCCESS)

C***                          Presentation-space handle into which the pointer is drawn
      INTEGER*4   HPS
C***                          X coordinate at which to draw the pointer, in device coordinates
      INTEGER*4   X
C***                          Y coordinate at which to draw the pointer, in device coordinates
      INTEGER*4   Y
C***                          Pointer handle
      INTEGER*4   POINTER
C***                          Shading control
      INTEGER*4   HALFTONE
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinDrawText - WIDTXT

```
      CALL WIDTXT (HPS, COUNT, TEXT, XLEFT, YBOTTOM, XRIGHT, YTOP, FORECOLOR, BACKCOLOR, CMD,
                   CHARS)

C***                                    Presentation-space handle
      INTEGER*4           HPS
C***                                    Count of the number of characters in the string
      INTEGER*4           COUNT
C***                                    Character string to be drawn
      CHARACTER*(count)   TEXT
C***                                    Text rectangle
      INTEGER*4           XLEFT
      INTEGER*4           YBOTTOM
      INTEGER*4           XRIGHT
      INTEGER*4           YTOP
C***                                    Foreground color
      INTEGER*4           FORECOLOR
C***                                    Background color
      INTEGER*4           BACKCOLOR
C***                                    An array of flags that determines how the text is drawn
      INTEGER*4           CMD
C***                                    Count of characters drawn within the rectangle
      INTEGER*4           CHARS
```

WinEmptyClipbrd - WIECB

```
CALL WIECB (HAB, SUCCESS)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Success indicator
      INTEGER*4   SUCCESS
```

WinEnablePhysInput - WDEPI

```
CALL WDEPI (DESKTOP, NEWINPUTSTATE, OLDINPUTSTATE)

C***                            Desktop-window handle
      INTEGER*4   DESKTOP
C***                            New state for the queuing of physical input
      INTEGER*4   NEWINPUTSTATE
C***                            Previous state for the queuing of physical input
      INTEGER*4   OLDINPUTSTATE
```

WinEnableWindow - WISENA

```
CALL WISENA (HWND, NEWENABLED, SUCCESS)

C***                          Window handle
      INTEGER*4   HWND
C***                          New enabled state
      INTEGER*4   NEWENABLED
C***                          Window enabled indicator
      INTEGER*4   SUCCESS
```

WinEnableWindowUpdate - WIEWU

```
CALL WIEWU (HWND, NEWVISIBILITY, SUCCESS)

C***                            Window handle
      INTEGER*4   HWND
C***                            New visibility state
      INTEGER*4   NEWVISIBILITY
C***                            Visibility-changed indicator
      INTEGER*4   SUCCESS
```

WinEndEnumWindows - WIEEW

```
CALL WIEEW (HENUM, SUCCESS)

C***                    Enumeration handle
      INTEGER*4   HENUM
C***                    Success indicator
      INTEGER*4   SUCCESS
```

WinEndPaint - WIEPNT

```
CALL WIEPNT (HPS, SUCCESS)

C***                     Presentation-space handle
      INTEGER*4   HPS
C***                     Success indicator
      INTEGER*4   SUCCESS
```

WinEnumClipbrdFmts - WIECF

```
      CALL WIECF (HAB, PREV, NEXT)

C***                  Anchor-block handle
      INTEGER*4   HAB
C***                  Previous clipboard-data format index
      INTEGER*4   PREV
C***                  Next clipboard-data format index
      INTEGER*4   NEXT
```

WinEnumDlgItem - WCEDLI

```
      CALL WCEDLI (DLG, HWND, CODE, LOCK, ITEM)

C***                  Dialog-window handle
      INTEGER*4   DLG
C***                  Child-window handle
      INTEGER*4   HWND
C***                  Item-type code
      INTEGER*4   CODE
C***                  Lock indicator
      INTEGER*4   LOCK
C***                  Item-window handle
      INTEGER*4   ITEM
```

WinEqualRect - WIEREC

```
      CALL WIEREC (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, XLEFT_2, YBOTTOM_2, XRIGHT_2, YTOP_2,
                   EQUAL)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        First rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Second rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                        Equality indicator
      INTEGER*4   EQUAL
```

WinExcludeUpdateRegion - WIEUPR

```
      CALL WIEUPR (HPS, HWND, COMPLEXITY)

C***                        Presentation-space handle whose clipping region is to be
                            updated
      INTEGER*4   HPS
C***                        Window handle
      INTEGER*4   HWND
C***                        Complexity value
      INTEGER*4   COMPLEXITY
```

WinFillRect - WIFRCT

```
      CALL WIFRCT (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, COLOR, SUCCESS)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Rectangle to be filled, in window coordinates
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Color with which to fill the rectangle
      INTEGER*4   COLOR
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinFindAtom - WMFNAT

```
      CALL WMFNAT (ATOMTBL, LENGTH, STRING, ATOM)

C***                                  Atom-table handle
      INTEGER*4           ATOMTBL
C***                                  Atom name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                  Atom value
      INTEGER*4           ATOM
```

WinFlashWindow - WIFLSH

```
      CALL WIFLSH (HWND, FLASH, SUCCESS)

C***                          Handle of window to be flashed
      INTEGER*4   HWND
C***                          Start-flashing indicator
      INTEGER*4   FLASH
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinFocusChange - WMFOCC

```
      CALL WMFOCC (DESKTOP, NEWFOCUS, FOCUSCHANGE, SUCCESS)

C***                             Desktop-window handle
      INTEGER*4   DESKTOP
C***                             Window handle to receive the focus
      INTEGER*4   NEWFOCUS
C***                             Focus changing indicators
      INTEGER*4   FOCUSCHANGE
C***                             Success indicator
      INTEGER*4   SUCCESS
```

WinFreeErrorInfo - WMFERI

```
      CALL WMFERI (ERRORINFO, SUCCESS)

C***                          Error-information block whose memory is to be released
                              (ERRINFO->HSTRUCT)
      INTEGER*4   ERRORINFO
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinFreeMsg - WUFMSG

```
      CALL WUFMSG (HAB, HWND, MSGID, PARAM1, PARAM2, REPLY, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window handle
      INTEGER*4   HWND
C***                        Message identity
      INTEGER*4   MSGID
C***                        Message parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                        Message parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                        Message reply (MRESULT: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   REPLY
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinGetClipPS - WPGCPS

```
      CALL WPGCPS (HWND, CLIPWINDOW, CLIPFLAGS, HPS)

C***                          Handle of window for which the presentation space is required
      INTEGER*4   HWND
C***                          Handle of window for clipping
      INTEGER*4   CLIPWINDOW
C***                          Clipping control flags
      INTEGER*4   CLIPFLAGS
C***                          Presentation-space handle that can be used for drawing
      INTEGER*4   HPS
```

WinGetCurrentTime - WIGCTI

```
      CALL WIGCTI (HAB, TIME)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    System-timer count
      INTEGER*4   TIME
```

WinGetDlgMsg - WCGETD

```
      CALL WCGETD (DLG, MSG, RESULT)

C***                                  Dialog-window handle
      INTEGER*4              DLG
C***                                  Message structure (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)   MSG
C***                                  Continue message indicator
      INTEGER*4              RESULT
```

WinGetErrorInfo - WMGEI

```
      CALL WMGEI (HAB, ERRORINFO)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Error information (ERRINFO→HSTRUCT)
      INTEGER*4   ERRORINFO
```

WinGetKeyState - WMKEYS

```
      CALL WMKEYS (DESKTOP, VK, KEYSTATE)

C***                        Desktop-window handle:
      INTEGER*4   DESKTOP
C***                        Virtual key value
      INTEGER*4   VK
C***                        Key state
      INTEGER*4   KEYSTATE
```

WinGetLastError - WMGLE

```
      CALL WMGLE (HAB, ERRORCODE)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Last-error state
      INTEGER*4   ERRORCODE
```

WinGetMinPosition - WIGMNP

```
      CALL WIGMNP (HWND, SWP, X, Y, SUCCESS)

C***                                Frame-window handle
      INTEGER*4            HWND
C***                                Set window position structure (EQUIVALENCEd onto
                                    SWP_BLOCK)
      CHARACTER*(SWP_LEN)  SWP
C***                                Preferred position
      INTEGER*4            X, Y
C***                                Success indicator
      INTEGER*4            SUCCESS
```

WinGetMsg - WMGET

```
      CALL WMGET (HAB, MSG, FILTER, FIRST, LAST, RESULT)

C***                              Anchor-block handle
      INTEGER*4             HAB
C***                              Message structure (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)  MSG
C***                              Window filter
      INTEGER*4             FILTER
C***                              First message identity
      INTEGER*4             FIRST
C***                              Last message identity
      INTEGER*4             LAST
C***                              Continue message indicator
      INTEGER*4             RESULT
```

WinGetNextWindow - WIGWIN

```
      CALL WIGWIN (HENUM, NEXT)

C***                    Enumeration handle
      INTEGER*4   HENUM
C***                    Next window handle in enumeration list
      INTEGER*4   NEXT
```

WinGetPhysKeyState - WMGPKS

```
CALL WMGPKS (DESKTOP, SCANCODE, KEYSTATE)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          Hardware scancode
      INTEGER*4   SCANCODE
C***                          Key state
      INTEGER*4   KEYSTATE
```

WinGetPS - WPGPS

```
CALL WPGPS (HWND, HPS)

C***                    Handle of window for which the presentation space is required
      INTEGER*4   HWND
C***                    Presentation-space handle that can be used for drawing in the window
      INTEGER*4   HPS
```

WinGetScreenPS - WPGSPS

```
CALL WPGSPS (DESKTOP, SCREENPS)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          Presentation-space handle
      INTEGER*4   SCREENPS
```

WinGetSysBitmap - WIQSBM

```
CALL WIQSBM (DESKTOP, INDEX, HBM)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          System bit-map index value
      INTEGER*4   INDEX
C***                          System bit-map handle
      INTEGER*4   HBM
```

WinInflateRect - WINFLR

```
CALL WINFLR (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, CX, CY, SUCCESS)

C***                         Anchor-block handle
      INTEGER*4   HAB
C***                         Rectangle to be expanded
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                         Horizontal expansion
      INTEGER*4   CX
C***                         Vertical expansion
      INTEGER*4   CY
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinInitialize - WMINIT

```
CALL WMINIT (OPTIONS, HAB)

C***                          Initialization options
      INTEGER*4   OPTIONS
C***                          Anchor-block handle
      INTEGER*4   HAB
```

WinInSendMsg - WCISMG

```
      CALL WCISMG (HAB, PROCESSING)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Message-processing indicator
      INTEGER*4   PROCESSING
```

WinInstStartApp - WMISAP

```
      CALL WMISAP (HINI, NOTIFYWINDOW, COUNT, APPLICATION, LENGTH, STRING, DATA, OPTIONS,
                   HAPP)

C***                                        Initialization-file handle
      INTEGER*4           HINI
C***                                        Notification-window handle
      INTEGER*4           NOTIFYWINDOW
C***                                        Count of elements in the Application parameter.
                                            Must be 1 or 2
      INTEGER*4           COUNT
C***                                        Identifier of the application to be started
                                            (STRL→HSTRUCT)
      INTEGER*4           APPLICATION(count)
C***                                        Input parameters for the application to be
                                            started
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                        Start data (STORAGE→HSTRUCT)
      INTEGER*4           DATA
C***                                        Option indicators
      INTEGER*4           OPTIONS
C***                                        Application handle
      INTEGER*4           HAPP
```

WinIntersectRect - WINTSR

```
      CALL WINTSR

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Intersection rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          First rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                          Second rectangle
      INTEGER*4   XLEFT_3
      INTEGER*4   YBOTTOM_3
      INTEGER*4   XRIGHT_3
      INTEGER*4   YTOP_3
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinInvalidateRect - WINVR

```
      CALL WINVR (HWND, XLEFT, YBOTTOM, XRIGHT, YTOP, INCLUDECLIPPEDCHILDREN, SUCCESS)

C***                                  Handle of window whose update region is to be
                                      changed
      INTEGER*4   HWND
C***                                  Update rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                                  Invalidation-scope indicator
      INTEGER*4   INCLUDECLIPPEDCHILDREN
C***                                  Success indicator
      INTEGER*4   SUCCESS
```

WinInvalidateRegion - WINVRG

```
      CALL WINVRG (HWND, HRGN, INCLUDECLIPPEDCHILDREN, SUCCESS)

C***                                  Handle of window whose update region is to be
                                      changed
      INTEGER*4   HWND
C***                                  Handle of the region to be added to the window's
                                      update region
      INTEGER*4   HRGN
C***                                  Invalidation-scope indicator
      INTEGER*4   INCLUDECLIPPEDCHILDREN
C***                                  Success indicator
      INTEGER*4   SUCCESS
```

WinInvertRect - WINVER

```
      CALL WINVER (HPS, XLEFT, YBOTTOM, XRIGHT, YTOP, SUCCESS)

C***                          Presentation-space handle
      INTEGER*4   HPS
C***                          Rectangle to be inverted
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinIsChild - WICHLD

```
      CALL WICHLD (CHILD, PARENT, RELATED)

C***                          Child-window handle
      INTEGER*4   CHILD
C***                          Parent-window handle
      INTEGER*4   PARENT
C***                          Related indicator
      INTEGER*4   RELATED
```

WinIsRectEmpty - WIREMP

```
      CALL WIREMP (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, EMPTY)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Empty indicator
      INTEGER*4   EMPTY
```

WinIsThreadActive - WMTHRA

```
      CALL WMTHRA (HAB, ACTIVE)

C***                        Anchor-block handle of calling thread
      INTEGER*4   HAB
C***                        Active-window indicator
      INTEGER*4   ACTIVE
```

WinIsWindow - WISWIN

```
      CALL WISWIN (HAB, HWND, VALID)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window handle
      INTEGER*4   HWND
C***                        Validity indicator
      INTEGER*4   VALID
```

WinIsWindowEnabled - WIQENA

```
      CALL WIQENA (HWND, ENABLED)

C***                        Window handle
      INTEGER*4   HWND
C***                        Enabled-state indicator
      INTEGER*4   ENABLED
```

WinIsWindowShowing - WIQSHO

```
      CALL WIQSHO (HWND, SHOWING)

C***                        Window handle
      INTEGER*4   HWND
C***                        Showing state indicator
      INTEGER*4   SHOWING
```

WinIsWindowVisible - WIQVIS

```
      CALL WIQVIS (HWND, VISIBLE)

C***                        Window handle
      INTEGER*4   HWND
C***                        Visibility-state indicator
      INTEGER*4   VISIBLE
```

WinLoadAccelTable - WMLACT

```
CALL WMLACT (HAB, RESOURCE, ACCELTABLE, ACCEL)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Resource identity containing the accelerator table
      INTEGER*4   RESOURCE
C***                          Accelerator-table identifier, within the resource file
      INTEGER*4   ACCELTABLE
C***                          Accelerator-table handle
      INTEGER*4   ACCEL
```

WinLoadDlg - WCLD

```
CALL WCLD (PARENT, OWNER, DLGPROC, RESOURCE, DLGID, CREATEPARAMS, DLG)

C***                          Parent-window handle of the created dialog window
      INTEGER*4   PARENT
C***                          Requested owner-window handle of the created dialog window
      INTEGER*4   OWNER
C***                          Dialog procedure for the created dialog window
      INTEGER*4   DLGPROC
C***                          Resource identity containing the dialog template
      INTEGER*4   RESOURCE
C***                          Dialog-template identity within the resource file
      INTEGER*4   DLGID
C***                          Application-defined data area (CREATEPARAMS→HSTRUCT)
      INTEGER*4   CREATEPARAMS
C***                          Dialog-window handle
      INTEGER*4   DLG
```

WinLoadLibrary - WMLLIB

```
CALL WMLLIB (HAB, LENGTH, STRING, LIBHANDLE)

C***                                Anchor-block handle
      INTEGER*4           HAB
C***                                Library name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                Library handle
      INTEGER*4           LIBHANDLE
```

WinLoadMenu - WCLM

```
CALL WCLM (OWNER, RESOURCE, MENUID, MENU)

C***                        Owner- and parent-window handle
      INTEGER*4   OWNER
C***                        Resource identifier
      INTEGER*4   RESOURCE
C***                        Menu identifier within the resource file
      INTEGER*4   MENUID
C***                        Menu-window handle
      INTEGER*4   MENU
```

WinLoadPointer - WMLPOI

```
CALL WMLPOI (DESKTOP, RESOURCE, POINTER, HPTR)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Resource identity containing the pointer definition
      INTEGER*4   RESOURCE
C***                        Identifier of the pointer to be loaded
      INTEGER*4   POINTER
C***                        Pointer handle
      INTEGER*4   HPTR
```

WinLoadProcedure - WMLPR

```
CALL WMLPR (HAB, LIBHANDLE, LENGTH, STRING, WNDPROC)

C***                                Anchor-block handle
      INTEGER*4           HAB
C***                                Library handle
      INTEGER*4           LIBHANDLE
C***                                Procedure name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                Window-procedure identifier
      INTEGER*4           WNDPROC
```

WinLoadString - WMLSTR

```
CALL WMLSTR (HAB, RESOURCE, STRING, BUFFERMAX, BUFFER, LENGTH)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Resource identity containing the string
      INTEGER*4             RESOURCE
C***                                    String identifier
      INTEGER*4             STRING
C***                                    Size of buffer
      INTEGER*4             BUFFERMAX
C***                                    Buffer that is to receive the string (LENGTH is
                                        result string length)
      CHARACTER*(buffermax) BUFFER
C***                                    The length of the string returned
      INTEGER*4             LENGTH
```

WinLockVisRegions - WILKVR

```
CALL WILKVR (DESKTOP, LOCK, SUCCESS)

C***                        Desktop-window handle or HWND_DESKTOP
      INTEGER*4   DESKTOP
C***                        Indicates whether the visible regions are being locked or
                            unlocked
      INTEGER*4   LOCK
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinLockWindow - WILKWN

```
CALL WILKWN (HWND, LOCK, RET)

C***                  Window handle
      INTEGER*4   HWND
C***                  Lock indicator
      INTEGER*4   LOCK
C***                  Locked-window handle
      INTEGER*4   RET
```

WinLockWindowUpdate - WMLKWU

```
CALL WMLKWU (DESKTOP, LOCKUPDATE, SUCCESS)

C***                          Desktop handle of the screen containing the window to be locked
     INTEGER*4   DESKTOP
C***                          Handle of window in which output is to be prevented
     INTEGER*4   LOCKUPDATE
C***                          Success indicator
     INTEGER*4   SUCCESS
```

WinMapDlgPoints - WCMDP

```
CALL WCMDP (DLG, X, Y, COUNT, OPTIONS, SUCCESS)

C***                                Dialog-window handle
     INTEGER*4   DLG
C***                                Coordinate points to be mapped
     INTEGER*4   X(count), Y(count)
C***                                Number of coordinate points
     INTEGER*4   COUNT
C***                                Calculation control
     INTEGER*4   OPTIONS
C***                                Coordinates-mapped indicator
     INTEGER*4   SUCCESS
```

WinMapWindowPoints - WIMAPW

```
CALL WIMAPW (FROM, TO, X, Y, COUNT, SUCCESS)

C***                                Handle of the window from whose coordinates points are
                                    to be mapped
     INTEGER*4   FROM
C***                                Handle of the window to whose coordinates points are to
                                    be mapped
     INTEGER*4   TO
C***                                Points to be mapped to the new coordinate system
     INTEGER*4   X(count), Y(count)
C***                                Number of points to be mapped
     INTEGER*4   COUNT
C***                                Success indicator
     INTEGER*4   SUCCESS
```

WinMessageBox - WCMBOX

```
CALL WCMBOX (PARENT, OWNER, LENGTH, STRING, LENGTH_2, STRING_2, WINDOW, STYLE,
             RESPONSE)

C***                                Parent-window handle of the created message-box window
     INTEGER*4            PARENT
C***                                Requested owner-window handle of the created
                                    message-box window
     INTEGER*4            OWNER
C***                                Message-box window message
     INTEGER*4            LENGTH
     CHARACTER*(length)   STRING
C***                                Message-box window title
     INTEGER*4            LENGTH_2
     CHARACTER*(length_2) STRING_2
C***                                Message-box window identity
     INTEGER*4            WINDOW
C***                                Message-box window style
     INTEGER*4            STYLE
C***                                User-response value
     INTEGER*4            RESPONSE
```

WinModifyDataStructure - WUMDS

```
CALL WUMDS (HAB, COUNT, TYPES, INLEN, INSTRUC, HANDLE, SUCCESS)

C***                                    Anchor-block handle
     INTEGER*4          HAB
C***                                    Number of elements
     INTEGER*4          COUNT
C***                                    Data type codes
     INTEGER*4          TYPES(count)
C***                                    Length of input structure
     INTEGER*4          INLEN
C***                                    Input structure
     CHARACTER*(inlen)  INSTRUC
C***                                    Handle of data structure to be modified
     INTEGER*4          HANDLE
C***                                    Success indicator
     INTEGER*4          SUCCESS
```

WinMsgMuxSemWait - WMMSWT

```
CALL WMMSWT (INDEXNBR, LIST, TIMEOUT, RC)

C***                        Index of the cleared semaphore in the list
     INTEGER*4   INDEXNBR
C***                        List of semaphore descriptors (STORAGE→HSTRUCT)
     INTEGER*4   LIST
C***                        Function time-out value
     INTEGER*4   TIMEOUT
C***                        Return code
     INTEGER*4   RC
```

WinMsgSemWait - WMSWT

```
CALL WMSWT (HSEM, TIMEOUT, RC)

C***                    Semaphore handle
     INTEGER*4   HSEM
C***                    Function time-out value
     INTEGER*4   TIMEOUT
C***                    Return code
     INTEGER*4   RC
```

WinMultWindowFromIDs - WIMWID

```
CALL WIMWID (PARENT, HWND, FIRST, LAST, WINDOWS)

C***                                    Parent-window handle
     INTEGER*4   PARENT
C***                                    Window handles
     INTEGER*4   HWND(last-first+1)
C***                                    First window identity value in the range (inclusive)
     INTEGER*4   FIRST
C***                                    Last window identity value in the range (inclusive)
     INTEGER*4   LAST
C***                                    Number of window handles returned
     INTEGER*4   WINDOWS
```

WinNextChar - WUNXTC

```
CALL WUNXTC (HAB, CODEPAGE, COUNTRY, LENGTH, STRING, CURRENTCHAR, NEXTCHAR)

C***                                    Anchor-block handle
     INTEGER*4           HAB
C***                                    Code page
     INTEGER*4           CODEPAGE
C***                                    Country code
     INTEGER*4           COUNTRY
C***                                    The string to be processed
     INTEGER*4           LENGTH
     CHARACTER*(length)  STRING
C***                                    Index of the current character in the string, in the
                                        range 1 (first character) to length of string (last
                                        character), or length+1 (end of string)
     INTEGER*4           CURRENTCHAR
C***                                    Index of the next character in the string. Length+1
                                        indicates that the end of the string has been reached
     INTEGER*4           NEXTCHAR
```

WinOffsetRect - WIOFFR

```
CALL WIOFFR (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, CX, CY, SUCCESS)

C***                        Anchor-block handle
     INTEGER*4   HAB
C***                        Rectangle to be offset
     INTEGER*4   XLEFT
     INTEGER*4   YBOTTOM
     INTEGER*4   XRIGHT
     INTEGER*4   YTOP
C***                        x value of offset
     INTEGER*4   CX
C***                        y value of offset
     INTEGER*4   CY
C***                        Success indicator
     INTEGER*4   SUCCESS
```

WinOpenClipbrd - WIOPCB

```
CALL WIOPCB (HAB, SUCCESS)

C***                        Anchor-block handle
     INTEGER*4   HAB
C***                        Success indicator
     INTEGER*4   SUCCESS
```

WinOpenWindowDC - WIOPDC

```
CALL WIOPDC (HWND, HDC)

C***                        Window handle
     INTEGER*4   HWND
C***                        Device-context handle
     INTEGER*4   HDC
```

WinPeekMsg - WMPEEK

```
      CALL WMPEEK (HAB, MSG, FILTER, FIRST, LAST, OPTIONS, RESULT)

C***                                  Anchor-block handle
      INTEGER*4             HAB
C***                                  Message structure (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)  MSG
C***                                  Window filter
      INTEGER*4             FILTER
C***                                  First message identity
      INTEGER*4             FIRST
C***                                  Last message identity
      INTEGER*4             LAST
C***                                  Options
      INTEGER*4             OPTIONS
C***                                  Message-available indicator
      INTEGER*4             RESULT
```

WinPostMsg - WMPOST

```
      CALL WMPOST (HWND, MSGID, PARAM1, PARAM2, RESULT)

C***                        Window handle
      INTEGER*4   HWND
C***                        Message identity
      INTEGER*4   MSGID
C***                        Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                        Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                        Message-posted indicator
      INTEGER*4   RESULT
```

WinPostQueueMsg - WCPQM

```
      CALL WCPQM (HMQ, MSGID, PARAM1, PARAM2, SUCCESS)

C***                        Message-queue handle
      INTEGER*4   HMQ
C***                        Message identifier
      INTEGER*4   MSGID
C***                        Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                        Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinPrevChar - WUPREC

```
CALL WUPREC (HAB, CODEPAGE, COUNTRY, LENGTH, STRING, CURRENTCHAR, PREVCHAR)

C***                                    Anchor-block handle
      INTEGER*4           HAB
C***                                    Code page
      INTEGER*4           CODEPAGE
C***                                    Country code
      INTEGER*4           COUNTRY
C***                                    The string to be processed
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                    Index of the current character in the string, in the
                                        range 1 (first character) to length of string (last
                                        character), or length+1 (end of string)
      INTEGER*4           CURRENTCHAR
C***                                    Index of the previous character in the string. 1
                                        indicates that the start of the string has been
                                        reached
      INTEGER*4           PREVCHAR
```

WinProcessDlg - WCPD

```
CALL WCPD (DLG, RESULT)

C***                  Dialog-window handle
      INTEGER*4   DLG
C***                  Reply value
      INTEGER*4   RESULT
```

WinPtInRect - WIPTRC

```
CALL WIPTRC (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, X, Y, F)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Rectangle to be queried
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                    Point to be queried
      INTEGER*4   X, Y
C***                    Success indicator
      INTEGER*4   F
```

WinQueryAccelTable - WMQACC

```
CALL WMQACC (HAB, FRAME, ACCEL)

C***                   Anchor-block handle
      INTEGER*4   HAB
C***                   Frame-window handle
      INTEGER*4   FRAME
C***                   Accelerator-table handle
      INTEGER*4   ACCEL
```

WinQueryActiveWindow - WMQACT

```
CALL WMQACT (PARENT, LOCK, ACTIVE)

C***                    Parent-window handle for which the active window is required
      INTEGER*4   PARENT
C***                    Windows lock-state indicator
      INTEGER*4   LOCK
C***                    Active-window handle
      INTEGER*4   ACTIVE
```

WinQueryAnchorBlock - WIQHAB

```
CALL WIQHAB (HWND, HAB)

C***                    Window handle
      INTEGER*4   HWND
C***                    Anchor block handle
      INTEGER*4   HAB
```

WinQueryAtomLength - WMQALN

```
CALL WMQALN (ATOMTBL, ATOM, RETLEN)

C***                        Atom-table handle
      INTEGER*4   ATOMTBL
C***                        Atom whose associated character-string length is to be returned
      INTEGER*4   ATOM
C***                        String length
      INTEGER*4   RETLEN
```

WinQueryAtomName - WMQANM

```
CALL WMQANM (ATOMTBL, ATOM, BUFFER, BUFFERMAX, RETLEN)

C***                                    Atom-table handle
      INTEGER*4              ATOMTBL
C***                                    Atom; identifies the character string to be retrieved
      INTEGER*4              ATOM
C***                                    Buffer to receive the character string (RETLEN is
                                        result string length)
      CHARACTER*(buffermax)  BUFFER
C***                                    Buffer size in bytes
      INTEGER*4              BUFFERMAX
C***                                    Length of retrieved character string
      INTEGER*4              RETLEN
```

WinQueryAtomUsage - WMQAUS

```
CALL WMQAUS (ATOMTBL, ATOM, COUNT)

C***                        Atom-table handle
      INTEGER*4   ATOMTBL
C***                        Atom whose use count is to be returned
      INTEGER*4   ATOM
C***                        Use count of the atom
      INTEGER*4   COUNT
```

WinQueryBits - WUQBIT

```
CALL WUQBIT (HAB, INT, DATA, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Integer whose bits are to be queried
      INTEGER*4   INT
C***                        Data indicating the status of the bits
      INTEGER*4   DATA(32)
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinQueryBitsUnderMask - WUQBM

```
CALL WUQBM (HAB, INT, MASK, BITVAL, SUCCESS)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Integer whose bits are to be queried
      INTEGER*4   INT
C***                    Mask identifying the bits to be queried
      INTEGER*4   MASK
C***                    Bit value extracted from the integer
      INTEGER*4   BITVAL
C***                    Success indicator
      INTEGER*4   SUCCESS
```

WinQueryCapture - WMQCAP

```
CALL WMQCAP (DESKTOP, LOCK, HWND)

C***                      Desktop-window handle
      INTEGER*4   DESKTOP
C***                      Lock-request indicator
      INTEGER*4   LOCK
C***                      Handle of the window with the pointer captured
      INTEGER*4   HWND
```

WinQueryClassInfo - WMQCLI

```
CALL WMQCLI (HAB, LENGTH, STRING, CLASSINFO, EXISTS)

C***                                    Anchor-block handle
      INTEGER*4                HAB
C***                                    Class name
      INTEGER*4                LENGTH
      CHARACTER*(length)       STRING
C***                                    Class information structure (EQUIVALENCEd onto
                                        CLASSINF_BLOCK)
      CHARACTER*(CLASSINF_LEN) CLASSINFO
C***                                    Class-exists indicator
      INTEGER*4                EXISTS
```

WinQueryClassName - WMQCLN

```
CALL WMQCLN (HWND, LENGTH, BUFFER, RETLEN)

C***                           Window handle
      INTEGER*4          HWND
C***                           Length of Buffer
      INTEGER*4          LENGTH
C***                           Class name (RETLEN is result string length)
      CHARACTER*(length) BUFFER
C***                           Returned class name length
      INTEGER*4          RETLEN
```

WinQueryClipbrdData - WIQCBD

```
CALL WIQCBD (HAB, FMT, DATA)

C***                Anchor-block handle
      INTEGER*4   HAB
C***                Format of the data to be accessed
      INTEGER*4   FMT
C***                Handle to the clipboard data
      INTEGER*4   DATA
```

WinQueryClipbrdFmtInfo - WIQCBF

```
CALL WIQCBF (HAB, FMT, FMTINFO, EXISTS)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Format of the data to be queried
      INTEGER*4   FMT
C***                    Memory model and usage flags
      INTEGER*4   FMTINFO
C***                    Format-exists indicator
      INTEGER*4   EXISTS
```

WinQueryClipbrdOwner - WIQCBO

```
CALL WIQCBO (HAB, LOCK, CLIPBRDOWNER)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Lock state indicator of clipboard owner window
      INTEGER*4   LOCK
C***                          Window handle of the current clipboard owner
      INTEGER*4   CLIPBRDOWNER
```

WinQueryClipbrdViewer - WIQCBV

```
CALL WIQCBV (HAB, LOCK, CLIPBRDVIEWER)

C***                           Anchor-block handle
      INTEGER*4   HAB
C***                           Lock-state indicator of clipboard viewer window
      INTEGER*4   LOCK
C***                           Current clipboard viewer window handle
      INTEGER*4   CLIPBRDVIEWER
```

WinQueryCp - WMGECP

```
CALL WMGECP (HMQ, CODEPAGE)

C***                     Message queue
      INTEGER*4   HMQ
C***                     Code page
      INTEGER*4   CODEPAGE
```

WinQueryCpList - WMQCPL

```
CALL WMQCPL (HAB, COUNT, CODEPAGE, TOTCOUNT)

C***                             Anchor-block handle
      INTEGER*4   HAB
C***                             Maximum number of code pages returned
      INTEGER*4   COUNT
C***                             Code page list
      INTEGER*4   CODEPAGE(count)
C***                             Total number of code pages available
      INTEGER*4   TOTCOUNT
```

WinQueryCursorInfo - WMQCUR

```
CALL WMQCUR (DESKTOP, CURSORINFO, CURSOR)

C***                                    Desktop-window handle
      INTEGER*4              DESKTOP
C***                                    Cursor information (EQUIVALENCEd onto CURINF_BLOCK)
      CHARACTER*(CURINF_LEN) CURSORINFO
C***                                    Current-cursor indicator
      INTEGER*4              CURSOR
```

WinQueryDataStructure - WUQDS

```
      CALL WUQDS (HAB, HANDLE, COUNT, TYPES, BUFLEN, OUTSTRUC, OUTLEN, SUCCESS)

C***                                        Anchor-block handle
      INTEGER*4           HAB
C***                                        Handle of data structure to be queried
      INTEGER*4           HANDLE
C***                                        Number of elements
      INTEGER*4           COUNT
C***                                        Data type codes
      INTEGER*4           TYPES(count)
C***                                        Length of buffer
      INTEGER*4           BUFLEN
C***                                        Output structure
      CHARACTER*(buflen)  OUTSTRUC
C***                                        Length of output structure
      INTEGER*4           OUTLEN
C***                                        Success indicator
      INTEGER*4           SUCCESS
```

WinQueryDefinition - WMQDEF

```
      CALL WMQDEF (HAB, PROGHANDLE, PROGRAMINFO, MAXLENGTH, LENGTH)

C***                              Anchor-block handle
      INTEGER*4   HAB
C***                              Handle of the program whose information is to be returned
      INTEGER*4   PROGHANDLE
C***                              Program-information data (PIBSTRUCT→HSTRUCT)
      INTEGER*4   PROGRAMINFO
C***                              Maximum length in bytes
      INTEGER*4   MAXLENGTH
C***                              Length of returned data
      INTEGER*4   LENGTH
```

WinQueryDesktopWindow - WIQDTW

```
      CALL WIQDTW (HAB, HDC, DESKTOP)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Device-context handle
      INTEGER*4   HDC
C***                        Desktop-window handle
      INTEGER*4   DESKTOP
```

WinQueryDlgItemShort - WCQDIS

```
      CALL WCQDIS (DLG, ITEM, RESULT, SIGNED, SUCCESS)

C***                        Parent-window handle
      INTEGER*4   DLG
C***                        Identity of the child window whose text is to be converted
      INTEGER*4   ITEM
C***                        Integer value resulting from the conversion
      INTEGER*4   RESULT
C***                        Sign indicator
      INTEGER*4   SIGNED
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinQueryDlgItemText - WMQDIT

```
CALL WMQDIT (DLG, ITEM, MAXTEXT, TEXT, RETLEN)

C***                                    Parent-window handle
      INTEGER*4            DLG
C***                                    Identity of the child window whose text is to be queried
      INTEGER*4            ITEM
C***                                    Length of Text
      INTEGER*4            MAXTEXT
C***                                    Output string (RETLEN is result string length)
      CHARACTER*(maxtext)  TEXT
C***                                    Actual number of characters returned
      INTEGER*4            RETLEN
```

WinQueryDlgItemTextLength - WMQDIL

```
CALL WMQDIL (DLG, ITEM, RETLEN)

C***                      Parent-window handle
      INTEGER*4   DLG
C***                      Identity of the child window whose text is to be queried
      INTEGER*4   ITEM
C***                      Length of text
      INTEGER*4   RETLEN
```

WinQueryFocus - WMQFOC

```
CALL WMQFOC (DESKTOP, LOCK, FOCUS)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Window lock state indicator
      INTEGER*4   LOCK
C***                        Focus-handle
      INTEGER*4   FOCUS
```

WinQueryMsgPos - WCQMPO

```
CALL WCQMPO (HAB, X, Y, SUCCESS)

C***                       Anchor-block handle
      INTEGER*4   HAB
C***                       Pointer position in screen coordinates
      INTEGER*4   X, Y
C***                       Success indicator
      INTEGER*4   SUCCESS
```

WinQueryMsgTime - WCQMTI

```
CALL WCQMTI (HAB, TIME)

C***                  Anchor-block handle
      INTEGER*4   HAB
C***                  Time in milliseconds
      INTEGER*4   TIME
```

WinQueryObjectWindow - WIQOBJ

```
CALL WIQOBJ (DESKTOP, OBJECT)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Object-window handle
      INTEGER*4   OBJECT
```

WinQueryPointer - WMQPTR

```
CALL WMQPTR (DESKTOP, POINTER)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          Pointer handle
      INTEGER*4   POINTER
```

WinQueryPointerInfo - WMQPTD

```
CALL WMQPTD (HPTR, POINTERINFO, SUCCESS)

C***                                      Pointer handle
      INTEGER*4               HPTR
C***                                      Pointer-information structure (EQUIVALENCEd onto
                                          PTRINF_BLOCK)
      CHARACTER*(PTRINF_LEN)  POINTERINFO
C***                                      Success indicator
      INTEGER*4               SUCCESS
```

WinQueryPointerPos - WMQPTP

```
CALL WMQPTP (DESKTOP, X, Y, SUCCESS)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          Pointer position in screen coordinates
      INTEGER*4   X, Y
C***                          Pointer position returned indicator
      INTEGER*4   SUCCESS
```

WinQueryPresParam - WMQPP

```
CALL WMQPP (HWND, ATTRTYPE1, ATTRTYPE2, ATTRTYPEFOUND, ATTRVALUELEN, ATTRVALUE,
            OPTIONS, RETLEN)

C***                             Window handle
      INTEGER*4   HWND
C***                             First attribute type identity
      INTEGER*4   ATTRTYPE1
C***                             Second attribute type identity
      INTEGER*4   ATTRTYPE2
C***                             Attribute type identity found
      INTEGER*4   ATTRTYPEFOUND
C***                             Byte count of the size of the AttrValue parameter
      INTEGER*4   ATTRVALUELEN
C***                             Attribute value (STORAGE→HSTRUCT)
      INTEGER*4   ATTRVALUE
C***                             Options
      INTEGER*4   OPTIONS
C***                             Length of presentation parameter value passed back
      INTEGER*4   RETLEN
```

WinQueryProfileData - WMQPDA

```
CALL WMQPDA (HAB, LENGTH, STRING, LENGTH_2, STRING_2, VALUE, SIZE, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Value data (STORAGE→HSTRUCT)
      INTEGER*4             VALUE
C***                                    Size of value data
      INTEGER*4             SIZE
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

WinQueryProfileInt - WMQPIN

```
CALL WMQPIN (HAB, LENGTH, STRING, LENGTH_2, STRING_2, DEFAULT, REQUIREDINTEGER)

C***                                        Anchor-block handle
      INTEGER*4             HAB
C***                                        Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                        Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                        Default value
      INTEGER*4             DEFAULT
C***                                        Key value specified in the intialization file
      INTEGER*4             REQUIREDINTEGER
```

WinQueryProfileSize - WMQPSZ

```
CALL WMQPSZ (HAB, LENGTH, STRING, LENGTH_2, STRING_2, VALUE, RETCODE)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Data length
      INTEGER*4             VALUE
C***                                    Success indicator
      INTEGER*4             RETCODE
```

WinQueryProfileString - WMQPST

```
CALL WMQPST (HAB, LENGTH, STRING, LENGTH_2, STRING_2, LENGTH_3, STRING_3,
             PROFILESTRING, MAXPSTRING, PSTRING)

C***                                        Anchor-block handle
      INTEGER*4             HAB
C***                                        Application name
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                        Key name
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                        Default string
      INTEGER*4             LENGTH_3
      CHARACTER*(length_3)  STRING_3
C***                                        Profile string (STORAGE→HSTRUCT)
      INTEGER*4             PROFILESTRING
C***                                        Maximum number
      INTEGER*4             MAXPSTRING
C***                                        Actual number
      INTEGER*4             PSTRING
```

WinQueryProgramTitles - WMQPTI

```
CALL WMQPTI (HAB, GROUP, BUFFER, BUFFERLEN, TOTAL, SUCCESS)

C***                                        Anchor-block handle
      INTEGER*4                 HAB
C***                                        Handle of the program group whose information
                                            is to be returned
      INTEGER*4                 GROUP
C***                                        Program information buffer (EQUIVALENCEd onto
                                            PROGENTR_BLOCK)
      CHARACTER*(PROGENTR_LEN)  BUFFER(total)
C***                                        Buffer length
      INTEGER*4                 BUFFERLEN
C***                                        Total number of structures
      INTEGER*4                 TOTAL
C***                                        Success indicator
      INTEGER*4                 SUCCESS
```

WinQueryQueueInfo - WMQQI

```
CALL WMQQI (HMQ, MQINFO, COPIED, SUCCESS)

C***                                Queue handle
      INTEGER*4             HMQ
C***                                Message queue information structure to contain the queue
                                    information (EQUIVALENCEd onto MQINF_BLOCK)
      CHARACTER*(MQINF_LEN) MQINFO
C***                                Size of message queue information structure that is
                                    provided (in bytes)
      INTEGER*4             COPIED
C***                                Success indicator
      INTEGER*4             SUCCESS
```

WinQueryQueueStatus - WCQSTA

```
CALL WCQSTA (DESKTOP, STATUS)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Status information
      INTEGER*4   STATUS
```

WinQuerySessionTitle - WMQSTI

```
CALL WMQSTI (HAB, SESSION, LENGTH, STRING, TITLELEN, RETCODE)

C***                                    Anchor-block handle
     INTEGER*4             HAB
C***                                    IBM Operating System/2 session identity of application
                                        whose title is requested
     INTEGER*4             SESSION
C***                                    Task title (LENGTH is result string length)
     INTEGER*4             LENGTH
     CHARACTER*(titlelen)  STRING
C***                                    Maximum length of data returnable in bytes
     INTEGER*4             TITLELEN
C***                                    Return code
     INTEGER*4             RETCODE
```

WinQuerySwitchEntry - WMQSEN

```
CALL WMQSEN (SWITCH, SWITCHDATA, RETCODE)

C***                                        Handle to the task list entry
     INTEGER*4               SWITCH
C***                                        Switch control data (EQUIVALENCEd onto
                                            SWCNTRL_BLOCK)
     CHARACTER*(SWCNTRL_LEN) SWITCHDATA
C***                                        Return code
     INTEGER*4               RETCODE
```

WinQuerySwitchHandle - WMQSHD

```
CALL WMQSHD (HWND, PROCESS, SWITCH)

C***                  Window handle of an application
     INTEGER*4  HWND
C***                  Process identity of the application
     INTEGER*4  PROCESS
C***                  Switch list handle for the specified application
     INTEGER*4  SWITCH
```

WinQuerySwitchList - WMQSLI

```
CALL WMQSLI (HAB, SWITCHENTRIES, DATALENGTH, COUNT)

C***                                      Anchor-block handle
     INTEGER*4               HAB
C***                                      Switch entries block (EQUIVALENCEd onto
                                          SWBLOCK_BLOCK)
     CHARACTER*(SWBLOCK_LEN) SWITCHENTRIES
C***                                      Maximum length of data returnable in bytes
     INTEGER*4               DATALENGTH
C***                                      Total number of switch list entries present in
                                          the system
     INTEGER*4               COUNT
```

WinQuerySysColor - WIQSCO

```
CALL WIQSCO (DESKTOP, COLOR, RESERVED, RGBCOLOR)

C***                  Desktop-window handle
     INTEGER*4  DESKTOP
C***                  System color-index value
     INTEGER*4  COLOR
C***                  Reserved
     INTEGER*4  RESERVED
C***                  RGB value
     INTEGER*4  RGBCOLOR
```

WinQuerySysModalWindow - WIQSMW

```
      CALL WIQSMW (DESKTOP, LOCK, SYSMODAL)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          Window's lock-state indicator
      INTEGER*4   LOCK
C***                          Handle of system modal window
      INTEGER*4   SYSMODAL
```

WinQuerySysPointer - WMQSPT

```
      CALL WMQSPT (DESKTOP, IDENTIFIER, COPY, POINTER)

C***                            Desktop-window handle
      INTEGER*4   DESKTOP
C***                            System-pointer identifier
      INTEGER*4   IDENTIFIER
C***                            Copy indicator
      INTEGER*4   COPY
C***                            Pointer handle
      INTEGER*4   POINTER
```

WinQuerySysValue - WMQSYS

```
      CALL WMQSYS (DESKTOP, VALUEID, VALUE)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          System-value identity
      INTEGER*4   VALUEID
C***                          System value
      INTEGER*4   VALUE
```

WinQuerySystemAtomTable - WMQSAT

```
      CALL WMQSAT (ATOMTBL)

C***                          System atom-table handle
      INTEGER*4   ATOMTBL
```

WinQueryTaskSizePos - WMQTSP

```
      CALL WMQTSP (HAB, SCREENGROUP, POSITIONDATA, RETCODE)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Screen group
      INTEGER*4             SCREENGROUP
C***                                    Window position and size data (EQUIVALENCEd onto
                                        SWP_BLOCK)
      CHARACTER*(SWP_LEN)   POSITIONDATA
C***                                    Return code
      INTEGER*4             RETCODE
```

WinQueryTaskTitle - WMQTTI

```
CALL WMQTTI (SESSION, LENGTH, STRING, TITLELEN, RETCODE)

C***                                  Session identity of application whose title is
                                      requested
      INTEGER*4             SESSION
C***                                  Task title (LENGTH is result string length)
      INTEGER*4             LENGTH
      CHARACTER*(titlelen)  STRING
C***                                  Maximum length of data returnable in bytes
      INTEGER*4             TITLELEN
C***                                  Return code
      INTEGER*4             RETCODE
```

WinQueryUpdateRect - WIQURC

```
CALL WIQURC (HWND, XLEFT, YBOTTOM, XRIGHT, YTOP, SUCCESS)

C***                        Handle of window whose update rectangle is to be queried
      INTEGER*4   HWND
C***                        Update region that bounds the rectangle (in window coordinates)
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinQueryUpdateRegion - WIQURG

```
CALL WIQURG (HWND, HRGN, COMPLEXITY)

C***                          Handle of window whose update region is to be queried
      INTEGER*4   HWND
C***                          Handle of the window's update region
      INTEGER*4   HRGN
C***                          Complexity of resulting region/error indicator
      INTEGER*4   COMPLEXITY
```

WinQueryValue - WUQVAL

```
CALL WUQVAL (HAB, VALUE, COUNT, TYPES, DATA, SUCCESS)

C***                            Anchor-block handle
      INTEGER*4   HAB
C***                            4-byte integer to be decomposed
      INTEGER*4   VALUE
C***                            Number of elements
      INTEGER*4   COUNT
C***                            Data types of the components of the value to be decomposed
      INTEGER*4   TYPES(count)
C***                            Data components extracted from the value to be decomposed
      INTEGER*4   DATA(count)
C***                            Success indicator
      INTEGER*4   SUCCESS
```

WinQueryVersion - WMQVER

```
CALL WMQVER (HAB, SYSINF)

C***                      Anchor-block handle
      INTEGER*4   HAB
C***                      System information within which the application is operating
      INTEGER*4   SYSINF
```

WinQueryWindow - WIQWIN

```
CALL WIQWIN (HWND, CODE, LOCK, RELATED)

C***                        Handle of window to query
      INTEGER*4   HWND
C***                        Type of window information
      INTEGER*4   CODE
C***                        Lock control
      INTEGER*4   LOCK
C***                        Window handle
      INTEGER*4   RELATED
```

WinQueryWindowDC - WIQWDC

```
CALL WIQWDC (HWND, HDC)

C***                        Window handle
      INTEGER*4   HWND
C***                        Device-context handle
      INTEGER*4   HDC
```

WinQueryWindowLockCount - WIQWLC

```
CALL WIQWLC (HWND, LOCK)

C***                        Window handle
      INTEGER*4   HWND
C***                        Window-lock count
      INTEGER*4   LOCK
```

WinQueryWindowPos - WIQPOS

```
CALL WIQPOS (HWND, SWP, SUCCESS)

C***                                  Window handle
      INTEGER*4            HWND
C***                                  SWP structure (EQUIVALENCEd onto SWP_BLOCK)
      CHARACTER*(SWP_LEN)  SWP
C***                                  Success indicator
      INTEGER*4            SUCCESS
```

WinQueryWindowProcess - WIQPRO

```
CALL WIQPRO (HWND, PID, TID, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Process identity of the thread that created the window
      INTEGER*4   PID
C***                        Thread identity of the thread that created the window
      INTEGER*4   TID
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinQueryWindowPtr - WIQWPT

```
CALL WIQWPT (HWND, B, P)

C***                        Window handle
      INTEGER*4   HWND
C***                        Index
      INTEGER*4   B
C***                        Pointer value (STORAGE->HSTRUCT)
      INTEGER*4   P
```

WinQueryWindowRect - WIQRCT

```
      CALL WIQRCT (HWND, XLEFT, YBOTTOM, XRIGHT, YTOP, SUCCESS)

C***                          Window handle
      INTEGER*4   HWND
C***                          Window rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          Rectangle-returned indicator
      INTEGER*4   SUCCESS
```

WinQueryWindowText - WIQTXT

```
      CALL WIQTXT (HWND, LENGTH, BUFFER, RETLEN)

C***                                  Window handle
      INTEGER*4           HWND
C***                                  Length
      INTEGER*4           LENGTH
C***                                  Window text (RETLEN is result string length)
      CHARACTER*(length)  BUFFER
C***                                  Length of returned text
      INTEGER*4           RETLEN
```

WinQueryWindowTextLength - WIQTXL

```
      CALL WIQTXL (HWND, RETLEN)

C***                        Window handle
      INTEGER*4   HWND
C***                        Length of the window text
      INTEGER*4   RETLEN
```

WinQueryWindowULong - WIQWUL

```
      CALL WIQWUL (HWND, B, VALUE)

C***                        Handle of window to be queried
      INTEGER*4   HWND
C***                        Index
      INTEGER*4   B
C***                        Value contained in the window word
      INTEGER*4   VALUE
```

WinQueryWindowUShort - WIQWUS

```
      CALL WIQWUS (HWND, B, VALUE)

C***                        Handle of window to be queried
      INTEGER*4   HWND
C***                        Index
      INTEGER*4   B
C***                        Value contained in the indicated window word
      INTEGER*4   VALUE
```

WinRegisterClass - WMRCL

```
CALL WMRCL (HAB, LENGTH, STRING, WNDPROC, CLASSSTYLE, EXTRA, REGISTERED)

C***                                    Anchor-block handle
      INTEGER*4           HAB
C***                                    Window-class name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                    Window-procedure identifier
      INTEGER*4           WNDPROC
C***                                    Default-window style
      INTEGER*4           CLASSSTYLE
C***                                    Reserved storage
      INTEGER*4           EXTRA
C***                                    Window-class-registration indicator
      INTEGER*4           REGISTERED
```

WinRegisterUserDatatype - WMRUD

```
CALL WMRUD (HAB, DATATYPE, COUNT, TYPES, SUCCESS)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Data type code to be defined
      INTEGER*4   DATATYPE
C***                          Number of elements
      INTEGER*4   COUNT
C***                          Data type codes of structure components
      INTEGER*4   TYPES(count)
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinRegisterUserMsg - WMRUM

```
CALL WMRUM (HAB, MSGID, TYPE1, DIR1, TYPE2, DIR2, TYPER, SUCCESS)

C***                    Anchor-block handle
      INTEGER*4   HAB
C***                    Message identifier
      INTEGER*4   MSGID
C***                    Datatype of message-parameter 1
      INTEGER*4   TYPE1
C***                    Direction of message-parameter 1
      INTEGER*4   DIR1
C***                    Datatype of message-parameter 2
      INTEGER*4   TYPE2
C***                    Direction of message-parameter 2
      INTEGER*4   DIR2
C***                    Datatype of message reply
      INTEGER*4   TYPER
C***                    Success indicator
      INTEGER*4   SUCCESS
```

WinRegisterWindowDestroy - WMREWD

```
CALL WMREWD (HWND, REGISTER, SUCCESS)

C***                      Window handle
      INTEGER*4   HWND
C***                      Registration indicator
      INTEGER*4   REGISTER
C***                      Success indicator
      INTEGER*4   SUCCESS
```

WinReleaseHook - WMRHK

```
      CALL WMRHK (HAB, HMQ, HOOK, ADDRESS, MODULE, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Handle of message queue from which the hook is to be released
      INTEGER*4   HMQ
C***                        Type of hook chain
      INTEGER*4   HOOK
C***                        Address of the hook routine
      INTEGER*4   ADDRESS
C***                        Module handle
      INTEGER*4   MODULE
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinReleasePS - WPREPS

```
      CALL WPREPS (HPS, SUCCESS)

C***                         Handle of the cache presentation space to release
      INTEGER*4   HPS
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinRemovePresParam - WMRPP

```
      CALL WMRPP (HWND, ATTRTYPE, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Attribute type identity
      INTEGER*4   ATTRTYPE
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinRemoveSwitchEntry - WMRSWE

```
      CALL WMRSWE (SWITCH, RETCODE)

C***                        Switch-list (task-list) entry handle
      INTEGER*4   SWITCH
C***                        Success indicator
      INTEGER*4   RETCODE
```

WinScrollWindow - WISCRL

```
      CALL WISCRL (HWND, DX, DY, XLEFT, YBOTTOM, XRIGHT, YTOP, XLEFT_2, YBOTTOM_2, XRIGHT_2,
                   YTOP_2, UPDATERGN, XLEFT_3, YBOTTOM_3, XRIGHT_3, YTOP_3, OPTIONS,
                   COMPLEXITY)

C***                                Window handle
      INTEGER*4   HWND
C***                                Amount of horizontal scroll to the right (in device units)
      INTEGER*4   DX
C***                                Amount of vertical scroll upward (in device units)
      INTEGER*4   DY
C***                                Scroll rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                                Clip rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                                Update region
      INTEGER*4   UPDATERGN
C***                                Update rectangle
      INTEGER*4   XLEFT_3
      INTEGER*4   YBOTTOM_3
      INTEGER*4   XRIGHT_3
      INTEGER*4   YTOP_3
C***                                Scroll options
      INTEGER*4   OPTIONS
C***                                Complexity of resulting region/error indicator
      INTEGER*4   COMPLEXITY
```

WinSendDlgItemMsg - WCSDIM

```
      CALL WCSDIM (DLG, ITEM, MSGID, PARAM1, PARAM2, REPLY)

C***                                Parent-window handle
      INTEGER*4   DLG
C***                                Identity of the child window
      INTEGER*4   ITEM
C***                                Message identity
      INTEGER*4   MSGID
C***                                Message parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                                Message parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                                Message-return data (MRESULT: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   REPLY
```

WinSendMsg - WMSEND

```
      CALL WMSEND (HWND, MSGID, PARAM1, PARAM2, REPLY)

C***                                Window handle
      INTEGER*4   HWND
C***                                Message identity
      INTEGER*4   MSGID
C***                                Parameter 1 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM1
C***                                Parameter 2 (MPARAM: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   PARAM2
C***                                Message-return data (MRESULT: Can also be HSTRUCT or COMPOSED)
      INTEGER*4   REPLY
```

WinSetAccelTable - WMSACC

```
      CALL WMSACC (HAB, ACCEL, FRAME, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Accelerator-table handle
      INTEGER*4   ACCEL
C***                        Frame-window handle
      INTEGER*4   FRAME
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetActiveWindow - WMSACT

```
      CALL WMSACT (DESKTOP, HWND, SUCCESS)

C***                         Desktop-window handle
      INTEGER*4   DESKTOP
C***                         Window handle
      INTEGER*4   HWND
C***                         Active-window-set indicator
      INTEGER*4   SUCCESS
```

WinSetBits - WUSBIT

```
      CALL WUSBIT (HAB, INT, DATA, SUCCESS)

C***                         Anchor-block handle
      INTEGER*4   HAB
C***                         Integer whose bits are to be modified
      INTEGER*4   INT
C***                         Data used to modify the integer
      INTEGER*4   DATA(32)
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinSetBitsUnderMask - WUSBM

```
      CALL WUSBM (HAB, INT, MASK, BITVAL, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Integer whose bits are to be modified
      INTEGER*4   INT
C***                        Mask identifying the bits to be modified
      INTEGER*4   MASK
C***                        Bit value to be used to modify the integer
      INTEGER*4   BITVAL
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetCapture - WMSCAP

```
      CALL WMSCAP (DESKTOP, HWND, SUCCESS)

C***                         Desktop-window handle, or HWND_DESKTOP
      INTEGER*4   DESKTOP
C***                         Handle of the window that is to receive all pointing device
                             messages
      INTEGER*4   HWND
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinSetClassMsgInterest - WMSCMI

```
CALL WMSCMI (HAB, LENGTH, STRING, MSGCLASS, CONTROL, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4           HAB
C***                                    Window-class name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                    Message class to have interest level set
      INTEGER*4           MSGCLASS
C***                                    Interest identifier for the message class
      INTEGER*4           CONTROL
C***                                    Interest-changed indicator
      INTEGER*4           SUCCESS
```

WinSetClipbrdData - WISCBD

```
CALL WISCBD (HAB, H, FMT, FMTINFO, DATAPLACED)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Handle
      INTEGER*4   H
C***                          Format
      INTEGER*4   FMT
C***                          Information
      INTEGER*4   FMTINFO
C***                          Data-placed indicator
      INTEGER*4   DATAPLACED
```

WinSetClipbrdOwner - WISCBO

```
CALL WISCBO (HAB, HWND, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window handle of the new clipboard owner
      INTEGER*4   HWND
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetClipbrdViewer - WISCBV

```
CALL WISCBV (HAB, NEWCLIPVIEWER, SUCCESS)

C***                              Anchor-block handle
      INTEGER*4   HAB
C***                              Window handle of the new clipboard viewer
      INTEGER*4   NEWCLIPVIEWER
C***                              Success indicator
      INTEGER*4   SUCCESS
```

WinSetCp - WMSECP

```
CALL WMSECP (HMQ, CODEPAGE, SUCCESS)

C***                         Message-queue handle
      INTEGER*4   HMQ
C***                         Code page
      INTEGER*4   CODEPAGE
C***                         Success indicator
      INTEGER*4   SUCCESS
```

WinSetDlgItemShort - WCSDI

```
CALL WCSDI (DLG, ITEM, VALUE, SIGNED, SUCCESS)

C***                    Parent-window handle
     INTEGER*4   DLG
C***                    Identity of the child window whose text is to be changed
     INTEGER*4   ITEM
C***                    Integer value used to generate the dialog item text
     INTEGER*4   VALUE
C***                    Sign indicator
     INTEGER*4   SIGNED
C***                    Success indicator
     INTEGER*4   SUCCESS
```

WinSetDlgItemText - WMSDIT

```
CALL WMSDIT (DLG, ITEM, LENGTH, STRING, SUCCESS)

C***                              Parent-window handle
     INTEGER*4           DLG
C***                              Identity of the child window whose text is to be set
     INTEGER*4           ITEM
C***                              Source string
     INTEGER*4           LENGTH
     CHARACTER*(length)  STRING
C***                              Success indicator
     INTEGER*4           SUCCESS
```

WinSetFocus - WMSFOC

```
CALL WMSFOC (DESKTOP, NEWFOCUS, SUCCESS)

C***                    Desktop-window handle
     INTEGER*4   DESKTOP
C***                    Window handle to receive the focus
     INTEGER*4   NEWFOCUS
C***                    Success indicator
     INTEGER*4   SUCCESS
```

WinSetHook - WMSHOO

```
CALL WMSHOO (HAB, HMQ, HOOKTYPE, HOOKPROC, MODULE, SUCCESS)

C***                    Anchor-block handle
     INTEGER*4   HAB
C***                    Queue identity
     INTEGER*4   HMQ
C***                    Hook chain type
     INTEGER*4   HOOKTYPE
C***                    Address of the application hook procedure
     INTEGER*4   HOOKPROC
C***                    Resource identity
     INTEGER*4   MODULE
C***                    Success indicator
     INTEGER*4   SUCCESS
```

WinSetKeyboardStateTable - WMSKST

```
CALL WMSKST (DESKTOP, KEYSTATETABLE, SET, SUCCESS)

C***                                  Desktop-window handle
      INTEGER*4     DESKTOP
C***                                  Key state table
      CHARACTER*1   KEYSTATETABLE(256)
C***                                  Set indicator
      INTEGER*4     SET
C***                                  Success indicator
      INTEGER*4     SUCCESS
```

WinSetMsgInterest - WMMINT

```
CALL WMMINT (HWND, MSGCLASS, CONTROL, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Message class to have interest level set
      INTEGER*4   MSGCLASS
C***                        Interest-identifier for the message class
      INTEGER*4   CONTROL
C***                        Interest-changed indicator
      INTEGER*4   SUCCESS
```

WinSetMsgMode - WMSMM

```
CALL WMSMM (HAB, LENGTH, STRING, CONTROL, SUCCESS)

C***                                Anchor block handle
      INTEGER*4           HAB
C***                                Window class name
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                                Message mode identifier
      INTEGER*4           CONTROL
C***                                Message delay indicator
      INTEGER*4           SUCCESS
```

WinSetMultWindowPos - WISMPS

```
CALL WISMPS (HAB, SWP, COUNT, SUCCESS)

C***                                  Anchor-block handle
      INTEGER*4           HAB
C***                                  Array (EQUIVALENCEd onto SWP_BLOCK)
      CHARACTER*(SWP_LEN)  SWP(count)
C***                                  Window count
      INTEGER*4           COUNT
C***                                  Positioning success indicator
      INTEGER*4           SUCCESS
```

WinSetOwner - WISOWN

```
CALL WISOWN (HWND, NEWOWNER, SUCCESS)

C***                        Window handle whose owner window is to be changed
      INTEGER*4   HWND
C***                        Handle of the new owner
      INTEGER*4   NEWOWNER
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetParent - WISPAR

```
      CALL WISPAR (HWND, NEWPARENT, REDRAW, SUCCESS)

C***                          Window handle
      INTEGER*4   HWND
C***                          New parent window handle
      INTEGER*4   NEWPARENT
C***                          Redraw indicator
      INTEGER*4   REDRAW
C***                          Parent-changed indicator
      INTEGER*4   SUCCESS
```

WinSetPointer - WMSPTR

```
      CALL WMSPTR (DESKTOP, NEWPOINTER, SUCCESS)

C***                           Desktop-window handle
      INTEGER*4   DESKTOP
C***                           New pointer handle
      INTEGER*4   NEWPOINTER
C***                           Pointer-updated indicator
      INTEGER*4   SUCCESS
```

WinSetPointerPos - WMSPTP

```
      CALL WMSPTP (DESKTOP, X, Y, SUCCESS)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          x position of pointer in screen coordinates
      INTEGER*4   X
C***                          y position of pointer in screen coordinates
      INTEGER*4   Y
C***                          Pointer position updated indicator
      INTEGER*4   SUCCESS
```

WinSetPresParam - WMSPP

```
      CALL WMSPP (HWND, ATTRTYPE, ATTRVALUELEN, ATTRVALUE, SUCCESS)

C***                            Window handle
      INTEGER*4   HWND
C***                            Attribute type identity
      INTEGER*4   ATTRTYPE
C***                            Byte count of the data passed in the AttrValue parameter
      INTEGER*4   ATTRVALUELEN
C***                            Attribute value (STORAGE→HSTRUCT)
      INTEGER*4   ATTRVALUE
C***                            Success indicator
      INTEGER*4   SUCCESS
```

WinSetRect - WISREC

```
CALL WISREC (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, LEFT, BOTTOM, RIGHT, TOP, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Rectangle to be updated
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Left edge of rectangle
      INTEGER*4   LEFT
C***                        Bottom edge of rectangle
      INTEGER*4   BOTTOM
C***                        Right edge of rectangle
      INTEGER*4   RIGHT
C***                        Top edge of rectangle
      INTEGER*4   TOP
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetRectEmpty - WISRE

```
CALL WISRE (HAB, XLEFT, YBOTTOM, XRIGHT, YTOP, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Rectangle to be set empty
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetSynchroMode - WMSSM

```
CALL WMSSM (HAB, MODE, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Synchronization mode
      INTEGER*4   MODE
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetSysColors - WISCOL

```
CALL WISCOL (DESKTOP, OPTIONS, FORMAT, START, TABLEN, TABLE, SUCCESS)

C***                        Desktop-window handle
      INTEGER*4   DESKTOP
C***                        Options
      INTEGER*4   OPTIONS
C***                        Format of entries in the table, as follows
      INTEGER*4   FORMAT
C***                        Starting system color index
      INTEGER*4   START
C***                        Number of elements
      INTEGER*4   TABLEN
C***                        Table
      INTEGER*4   TABLE(tablen)
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetSysModalWindow - WISSMW

```
      CALL WISSMW (DESKTOP, HWND, SUCCESS)

C***                          Desktop-window handle, or HWND_DESKTOP
      INTEGER*4   DESKTOP
C***                          Handle of window to become system-modal window
      INTEGER*4   HWND
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinSetSysValue - WMSSYS

```
      CALL WMSSYS (DESKTOP, VALUEID, VALUE, SUCCESS)

C***                          Desktop-window handle
      INTEGER*4   DESKTOP
C***                          System-value identity
      INTEGER*4   VALUEID
C***                          System value
      INTEGER*4   VALUE
C***                          Value-set indicator
      INTEGER*4   SUCCESS
```

WinSetValue - WUSVAL

```
      CALL WUSVAL (HAB, COUNT, TYPES, DATA, VALUE, SUCCESS)

C***                              Anchor-block handle
      INTEGER*4   HAB
C***                              Number of elements
      INTEGER*4   COUNT
C***                              Data types of the components of the value to be composed
      INTEGER*4   TYPES(count)
C***                              Data components of the value to be composed
      INTEGER*4   DATA(count)
C***                              4-byte integer
      INTEGER*4   VALUE
C***                              Success indicator
      INTEGER*4   SUCCESS
```

WinSetWindowBits - WISWBT

```
      CALL WISWBT (HWND, B, DATA, MASK, SUCCESS)

C***                          Window handle
      INTEGER*4   HWND
C***                          Zero-based index of the value to be set
      INTEGER*4   B
C***                          Bit data to store in the window words
      INTEGER*4   DATA
C***                          Bits to be written indicator
      INTEGER*4   MASK
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinSetWindowPos - WISPOS

```
CALL WISPOS (HWND, BEHIND, X, Y, CX, CY, OPTIONS, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Relative window-placement order
      INTEGER*4   BEHIND
C***                        Window position, x coordinate
      INTEGER*4   X
C***                        Window position, y coordinate
      INTEGER*4   Y
C***                        Window size
      INTEGER*4   CX
C***                        Window size
      INTEGER*4   CY
C***                        Window-positioning options
      INTEGER*4   OPTIONS
C***                        Repositioning indicator
      INTEGER*4   SUCCESS
```

WinSetWindowPtr - WISWPT

```
CALL WISWPT (HWND, B, P, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Zero-based index into the window words
      INTEGER*4   B
C***                        Pointer value to store in the window words (STORAGE→HSTRUCT)
      INTEGER*4   P
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetWindowText - WISTXT

```
CALL WISTXT (HWND, LENGTH, STRING, RESULT)

C***                              Window handle
      INTEGER*4           HWND
C***                              Window text
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
C***                              Text-updated indicator
      INTEGER*4           RESULT
```

WinSetWindowULong - WISWUL

```
CALL WISWUL (HWND, B, DATA, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Zero-based index of the value to be set
      INTEGER*4   B
C***                        Unsigned, long integer value to store in the window words
      INTEGER*4   DATA
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSetWindowUShort - WISWUS

```
      CALL WISWUS (HWND, B, DATA, SUCCESS)

C***                        Window handle
      INTEGER*4   HWND
C***                        Zero-based index of the value to be set
      INTEGER*4   B
C***                        Unsigned, short integer value to store in the window words
      INTEGER*4   DATA
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinShowCursor - WISCUR

```
      CALL WISCUR (HWND, SHOW, SUCCESS)

C***                        Handle of window to which the cursor belongs
      INTEGER*4   HWND
C***                        Show indicator
      INTEGER*4   SHOW
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinShowPointer - WMSHPT

```
      CALL WMSHPT (DESKTOP, SHOW, SUCCESS)

C***                         Desktop-window handle
      INTEGER*4   DESKTOP
C***                         Level-update indicator
      INTEGER*4   SHOW
C***                         Display-level-updated indicator
      INTEGER*4   SUCCESS
```

WinShowTrackRect - WISTR

```
      CALL WISTR (HWND, SHOW, SUCCESS)

C***                       Window handle
      INTEGER*4   HWND
C***                       Show indicator
      INTEGER*4   SHOW
C***                       Success indicator
      INTEGER*4   SUCCESS
```

WinShowWindow - WISHOW

```
      CALL WISHOW (HWND, NEWVISIBILITY, SUCCESS)

C***                             Window handle
      INTEGER*4   HWND
C***                             New visibility state
      INTEGER*4   NEWVISIBILITY
C***                             Visibility changed indicator
      INTEGER*4   SUCCESS
```

WinStartTimer - WISTAT

```
CALL WISTAT (HAB, HWND, TIMER, TIMEOUT, RET)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window handle that is part of the timer identification
      INTEGER*4   HWND
C***                        Timer identifier
      INTEGER*4   TIMER
C***                        Delay time in milliseconds
      INTEGER*4   TIMEOUT
C***                        Return code
      INTEGER*4   RET
```

WinStopTimer - WISTOT

```
CALL WISTOT (HAB, HWND, TIMER, SUCCESS)

C***                        Anchor-block handle
      INTEGER*4   HAB
C***                        Window handle
      INTEGER*4   HWND
C***                        Timer identifier
      INTEGER*4   TIMER
C***                        Success indicator
      INTEGER*4   SUCCESS
```

WinSubstituteStrings - WUSUBS

```
CALL WUSUBS (HWND, LENGTH, STRING, DESTMAX, DEST, DESTRET)

C***                                Handle of window that processes the call
      INTEGER*4            HWND
C***                                Source string
      INTEGER*4            LENGTH
      CHARACTER*(length)   STRING
C***                                Maximum number of characters returnable
      INTEGER*4            DESTMAX
C***                                Resultant string (DESTRET is result string length)
      CHARACTER*(destmax)  DEST
C***                                Actual number of characters returned
      INTEGER*4            DESTRET
```

WinSubtractRect - WISUBR

```
CALL WISUBR

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Result
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          First source rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                          Second source rectangle
      INTEGER*4   XLEFT_3
      INTEGER*4   YBOTTOM_3
      INTEGER*4   XRIGHT_3
      INTEGER*4   YTOP_3
C***                          Not-empty indicator
      INTEGER*4   NONEMPTY
```

WinSwitchToProgram - WMSTP

```
      CALL WMSTP (SWHANDLE, RETCODE)

C***                        Switch list entry handle of program to be activated
      INTEGER*4   SWHANDLE
C***                        Return code
      INTEGER*4   RETCODE
```

WinTerminate - WMTERM

```
      CALL WMTERM (HAB, TERMINATED)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Termination indicator
      INTEGER*4   TERMINATED
```

WinTerminateApp - WMSTOP

```
      CALL WMSTOP (HAPP, SUCCESS)

C***                       Application handle
      INTEGER*4   HAPP
C***                       Success indicator
      INTEGER*4   SUCCESS
```

WinTrackRect - WITREC

```
      CALL WITREC (HWND, HPS, TRACKINFO, SUCCESS)

C***                                      Window handle where tracking is to take place
      INTEGER*4                 HWND
C***                                      Presentation-space handle
      INTEGER*4                 HPS
C***                                      Track information (EQUIVALENCEd onto
                                          TRACKINF_BLOCK)
      CHARACTER*(TRACKINF_LEN)  TRACKINFO
C***                                      Success indicator
      INTEGER*4                 SUCCESS
```

WinTranslateAccel - WMTAC

```
      CALL WMTAC (HAB, HWND, ACCEL, QMSG, SUCCESS)

C***                               Anchor-block handle
      INTEGER*4             HAB
C***                               Destination window
      INTEGER*4             HWND
C***                               Accelerator-table handle
      INTEGER*4             ACCEL
C***                               Message to be translated (EQUIVALENCEd onto QMSG_BLOCK)
      CHARACTER*(QMSG_LEN)  QMSG
C***                               Success indicator
      INTEGER*4             SUCCESS
```

WinUnionRect - WIUNRC

```
      CALL WIUNRC

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Bounding rectangle
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                          First source rectangle
      INTEGER*4   XLEFT_2
      INTEGER*4   YBOTTOM_2
      INTEGER*4   XRIGHT_2
      INTEGER*4   YTOP_2
C***                          Second source rectangle
      INTEGER*4   XLEFT_3
      INTEGER*4   YBOTTOM_3
      INTEGER*4   XRIGHT_3
      INTEGER*4   YTOP_3
C***                          Non-empty indicator
      INTEGER*4   NONEMPTY
```

WinUpdateWindow - WIUPDT

```
      CALL WIUPDT (HWND, SUCCESS)

C***                          Window handle
      INTEGER*4   HWND
C***                          Window-updated indicator
      INTEGER*4   SUCCESS
```

WinUpper - WUPPER

```
      CALL WUPPER (HAB, CODEPAGE, COUNTRY, LENGTH, STRING)

C***                                  Anchor-block handle
      INTEGER*4           HAB
C***                                  Code page
      INTEGER*4           CODEPAGE
C***                                  Country code
      INTEGER*4           COUNTRY
C***                                  String to be converted to uppercase
      INTEGER*4           LENGTH
      CHARACTER*(length)  STRING
```

WinUpperChar - WUPCH

```
      CALL WUPCH (HAB, CODEPAGE, COUNTRY, INCHAR, OUTCHAR)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          Code page
      INTEGER*4   CODEPAGE
C***                          Country code
      INTEGER*4   COUNTRY
C***                          Character to be translated to uppercase
      INTEGER*4   INCHAR
C***                          Translated character
      INTEGER*4   OUTCHAR
```

WinValidateRect - WIVREC

```
      CALL WIVREC (HWND, XLEFT, YBOTTOM, XRIGHT, YTOP, INCLUDECLIPPEDCHILDREN, SUCCESS)

C***                                  Handle of window whose update region is changed
      INTEGER*4   HWND
C***                                  Rectangle to be subtracted from the window's update
                                      region
      INTEGER*4   XLEFT
      INTEGER*4   YBOTTOM
      INTEGER*4   XRIGHT
      INTEGER*4   YTOP
C***                                  Validation-scope indicator
      INTEGER*4   INCLUDECLIPPEDCHILDREN
C***                                  Success indicator
      INTEGER*4   SUCCESS
```

WinValidateRegion - WIVRGN

```
      CALL WIVRGN (HWND, HRGN, INCLUDECLIPPEDCHILDREN, SUCCESS)

C***                                  Handle of window whose update region is changed
      INTEGER*4   HWND
C***                                  Handle of subtracted region
      INTEGER*4   HRGN
C***                                  Validation-scope indicator
      INTEGER*4   INCLUDECLIPPEDCHILDREN
C***                                  Success indicator
      INTEGER*4   SUCCESS
```

WinWaitMsg - WMWAIT

```
      CALL WMWAIT (HAB, FIRST, LAST, SUCCESS)

C***                          Anchor-block handle
      INTEGER*4   HAB
C***                          First message identity
      INTEGER*4   FIRST
C***                          Last message identity
      INTEGER*4   LAST
C***                          Success indicator
      INTEGER*4   SUCCESS
```

WinWindowFromDC - WIQHDC

```
      CALL WIQHDC (HDC, HWND)

C***                       Device-context handle
      INTEGER*4   HDC
C***                       Window handle
      INTEGER*4   HWND
```

WinWindowFromID - WIQHID

```
      CALL WIQHID (PARENT, IDENTIFIER, HWND)

C***                            Parent-window handle
      INTEGER*4   PARENT
C***                            Identity of the child window
      INTEGER*4   IDENTIFIER
C***                            Window handle
      INTEGER*4   HWND
```

WinWindowFromPoint - WIWFP

```
      CALL WIWFP (PARENT, X, Y, ENUMCHILDREN, LOCK, FOUND)

C***                              Window handle whose child windows are to be tested
      INTEGER*4   PARENT
C***                              The point to be tested
      INTEGER*4   X, Y
C***                              Test control
      INTEGER*4   ENUMCHILDREN
C***                              Lock control
      INTEGER*4   LOCK
C***                              Window handle beneath Point
      INTEGER*4   FOUND
```

WinWriteProfileData - WMWPDA

```
      CALL WMWPDA (HAB, LENGTH, STRING, LENGTH_2, STRING_2, VALUE, SIZE, SUCCESS)

C***                                    Anchor-block handle
      INTEGER*4             HAB
C***                                    Text string
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Text string
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Value data (STORAGE→HSTRUCT)
      INTEGER*4             VALUE
C***                                    Size of value data
      INTEGER*4             SIZE
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

WinWriteProfileString - WMWPST

```
      CALL WMWPST (HAB, LENGTH, STRING, LENGTH_2, STRING_2, LENGTH_3, STRING_3, SUCCESS)

C***                                    Anchor block handle
      INTEGER*4             HAB
C***                                    Text string
      INTEGER*4             LENGTH
      CHARACTER*(length)    STRING
C***                                    Text string
      INTEGER*4             LENGTH_2
      CHARACTER*(length_2)  STRING_2
C***                                    Text string
      INTEGER*4             LENGTH_3
      CHARACTER*(length_3)  STRING_3
C***                                    Success indicator
      INTEGER*4             SUCCESS
```

Index

A

ACCEL 2-1
ACCELTABLE 2-1
ARCPARAM 2-1
ARCPARAMS 2-1
AREABUNDLE 2-1
ATOM 2-2

B

BANDRECT 2-2
bit utilities 1-4
BITMAPINFO 2-2
BITMAPINFOHEADER 2-2
BIT16 2-3
BIT32 2-3
BIT8 2-2
BOOL 2-3
BTNCDATA 2-3
BUFFER 2-3
BUNDLE 2-3
BYTE 2-3

C

CHAR 2-3
CHARBUNDLE 2-4
CLASSINFO 2-4
COMPOSED 2-4
composed-integer utilities 1-4
COUNT2 2-4
COUNT2B 2-4
COUNT2CH 2-4
COUNT4 2-4
COUNT4B 2-4
CPID 2-5
CREATEPARAMS 2-5
CREATESTRUCT 2-5
CTLDATA 2-5
CURSORINFO 2-6

D

DevCloseDC 3-1
DevEscape 3-1
DEVOPENDATA 2-6
DevOpenDC 3-1
DEVOPENSTRUC 2-6
DevPostDeviceModes 3-2
DevQueryCaps 3-2
DevQueryDeviceNames 3-2
DevQueryHardcopyCaps 3-3
DLGTEMPLATE 2-7
DLGTITEM 2-7
DRIVDATA 2-7

E

ENTRYFDATA 2-8
ERRINFO 2-8
ERRORID 2-8

F

FATTRS 2-8
FFDESCS 2-8
FIXED 2-9
FONTMETRICS 2-9
FORTRAN coding techniques
- data structures 1-1
- datatype mapping 1-3
- FALSE (FORTRAN) 1-3
- interdependent parameters 1-3
- maximum name length 1-3
- notation conventions 1-3
- NULL (FORTRAN) 1-3
- TRUE (FORTRAN) 1-3

FRAMECDATA 2-11

G

GIATR 4-33
GIBB 4-1
GIBOX 4-2
GICA 4-35
GICALM 4-2
GICB 4-35
GICH 4-35
GICHAR 4-2
GICHPP 4-3
GICHPR 4-3
GICONV 4-4
GICORF 4-5
GICORS 4-5
GICP 4-36
GICSEG 4-6
GIDAP 4-36
GIDATR 4-36
GIECR 4-10
GIFARC 4-10
GIFLD 4-38
GIGARC 4-14
GIGLW 4-39
GIICR 4-11
GILINE 4-11
GILW 4-39
GIMARK 4-12
GIMB 4-39
GIMOVE 4-13
GIMRKS 4-15
GIOCR 4-13
GIOREG 4-13
GIPAP 4-41
GIPARC 4-14
GIPCRT 4-7
GIPDEL 4-8
GIPEL 4-40
GIPFLS 4-15
GIPFLT 4-14
GIPIR 4-15
GIPLNE 4-15
GIPQRY 4-27

GIPRP 4-40
GIPS 4-41
GIPSPL 4-15
GIPV 4-40
GIPVIS 4-16
GIQAP 4-16
GIQATR 4-16
GIQBD 4-17
GIQCA 4-18
GIQCB 4-18
GIQCEL 4-21
GIQCH 4-18
GIQCLB 4-19
GIQCP 4-20
GIQCPO 4-19
GIQCRO 4-19
GIQCTM 4-26
GIQDAP 4-20
GIQDAT 4-21
GIQDVL 4-21
GIQDVM 4-21
GIQFLD 4-23
GIQGLW 4-24
GIQLW 4-24
GIQMB 4-25
GIQPAP 4-27
GIQPAS 4-27
GIQPEL 4-27
GIQPRP 4-27
GIQPV 4-26
GIQRB 4-28
GIQRR 4-28
GIQSVL 4-31
GIQTB 4-30
GIQTFM 4-30
GIQVTM 4-31
GIREG 4-7
GIREGN 4-41
GIRIR 4-31
GIROT 4-32
GIRVIS 4-32
GISAP 4-33
GISCAL 4-33
GISCTM 4-40
GISDVL 4-37
GISDVM 4-37
GISPAS 4-41
GISTFM 4-42
GISVL 4-42
GISVTM 4-43
GIWBB 4-44
GIXLAT 4-43
GpiAssociate 4-1
GpiBeginArea 4-1
GpiBeginElement 4-1
GpiBeginPath 4-1
GpiBitBlt 4-1
GpiBox 4-2
GpiCallSegmentMatrix 4-2
GpiCharString 4-2
GpiCharStringAt 4-2
GpiCharStringPos 4-3
GpiCharStringPosAt 4-3
GpiCloseFigure 4-3
GpiCloseSegment 4-3
GpiCombineRegion 4-4
GpiComment 4-4
GpiConvert 4-4
GpiCopyMetaFile 4-4
GpiCorrelateChain 4-5
GpiCorrelateFrom 4-5
GpiCorrelateSegment 4-6
GpiCreateBitmap 4-6
GpiCreateLogColorTable 4-6
GpiCreateLogFont 4-7
GpiCreatePS 4-7
GpiCreateRegion 4-7
GpiDeleteBitmap 4-7
GpiDeleteElement 4-7
GpiDeleteElementRange 4-8
GpiDeleteElementsBetweenLabels 4-8
GpiDeleteMetaFile 4-8
GpiDeleteSegment 4-8
GpiDeleteSegments 4-8
GpiDeleteSetId 4-8
GpiDestroyPS 4-8
GpiDestroyRegion 4-9
GpiDrawChain 4-9
GpiDrawDynamics 4-9
GpiDrawFrom 4-9
GpiDrawSegment 4-9
GpiEndArea 4-9
GpiEndElement 4-9
GpiEndPath 4-9
GpiEqualRegion 4-10
GpiErase 4-10
GpiErrorSegmentData 4-10
GpiExcludeClipRectangle 4-10
GpiFillPath 4-10
GpiFullArc 4-10
GpiGetData 4-11
GpiImage 4-11
GpiIntersectClipRectangle 4-11
GpiLabel 4-11
GpiLine 4-11
GpiLoadBitmap 4-12
GpiLoadFonts 4-12
GpiLoadMetaFile 4-12
GpiMarker 4-12
GpiModifyPath 4-12
GpiMove 4-13
GpiOffsetClipRegion 4-13
GpiOffsetElementPointer 4-13
GpiOffsetRegion 4-13
GpiOpenSegment 4-13
GpiOutlinePath 4-13
GpiPaintRegion 4-13
GpiPartialArc 4-14
GpiPlayMetaFile 4-14
GpiPointArc 4-14
GpiPolyFillet 4-14
GpiPolyFilletSharp 4-15
GpiPolyLine 4-15
GpiPolyMarker 4-15
GpiPolySpline 4-15
GpiPop 4-15
GpiPtInRegion 4-15
GpiPtVisible 4-16
GpiPutData 4-16
GpiQueryArcParams 4-16
GpiQueryAttrMode 4-16
GpiQueryAttrs 4-16
GpiQueryBackColor 4-16
GpiQueryBackMix 4-17

GpiQueryBitmapBits 4-17
GpiQueryBitmapDimension 4-17
GpiQueryBitmapHandle 4-17
GpiQueryBitmapParameters 4-17
GpiQueryBoundaryData 4-17
GpiQueryCharAngle 4-18
GpiQueryCharBox 4-18
GpiQueryCharDirection 4-18
GpiQueryCharMode 4-18
GpiQueryCharSet 4-18
GpiQueryCharShear 4-18
GpiQueryCharStringPos 4-19
GpiQueryCharStringPosAt 4-19
GpiQueryClipBox 4-19
GpiQueryClipRegion 4-19
GpiQueryColor 4-20
GpiQueryColorData 4-20
GpiQueryColorIndex 4-20
GpiQueryCp 4-20
GpiQueryCurrentPosition 4-20
GpiQueryDefArcParams 4-20
GpiQueryDefAttrs 4-21
GpiQueryDefaultViewMatrix 4-21
GpiQueryDefCharBox 4-21
GpiQueryDefTag 4-21
GpiQueryDefViewingLimits 4-21
GpiQueryDevice 4-21
GpiQueryDeviceBitmapFormats 4-22
GpiQueryDrawControl 4-22
GpiQueryDrawingMode 4-22
GpiQueryEditMode 4-22
GpiQueryElementPointer 4-22
GpiQueryElementType 4-22
GpiQueryFontFileDescriptions 4-23
GpiQueryFontMetrics 4-23
GpiQueryFonts 4-23
GpiQueryGraphicsField 4-23
GpiQueryInitialSegmentAttrs 4-24
GpiQueryKerningPairs 4-24
GpiQueryLineEnd 4-24
GpiQueryLineJoin 4-24
GpiQueryLineType 4-24
GpiQueryLineWidth 4-24
GpiQueryLineWidthGeom 4-24
GpiQueryLogColorTable 4-25
GpiQueryMarker 4-25
GpiQueryMarkerBox 4-25
GpiQueryMarkerSet 4-25
GpiQueryMetaFileBits 4-25
GpiQueryMetaFileLength 4-25
GpiQueryMix 4-26
GpiQueryModelTransformMatrix 4-26
GpiQueryNearestColor 4-26
GpiQueryNumberSetIds 4-26
GpiQueryPageViewport 4-26
GpiQueryPattern 4-26
GpiQueryPatternRefPoint 4-27
GpiQueryPatternSet 4-27
GpiQueryPel 4-27
GpiQueryPickAperturePosition 4-27
GpiQueryPickApertureSize 4-27
GpiQueryPS 4-27
GpiQueryRealColors 4-28
GpiQueryRegionBox 4-28
GpiQueryRegionRects 4-28
GpiQueryRGBColor 4-29
GpiQuerySegmentAttrs 4-29
GpiQuerySegmentNames 4-29
GpiQuerySegmentPriority 4-29
GpiQuerySegmentTransformMatrix 4-30
GpiQuerySetIds 4-30
GpiQueryStopDraw 4-30
GpiQueryTag 4-30
GpiQueryTextBox 4-30
GpiQueryViewingLimits 4-31
GpiQueryViewingTransformMatrix 4-31
GpiQueryWidthTable 4-31
GpiRealizeColorTable 4-31
GpiRectInRegion 4-31
GpiRectVisible 4-32
GpiRemoveDynamics 4-32
GpiResetBoundaryData 4-32
GpiResetPS 4-32
GpiRestorePS 4-32
GpiRotate 4-32
GpiSaveMetaFile 4-33
GpiSavePS 4-33
GpiScale 4-33
GpiSetArcParams 4-33
GpiSetAttrMode 4-33
GpiSetAttrs 4-33
GpiSetBackColor 4-34
GpiSetBackMix 4-34
GpiSetBitmap 4-34
GpiSetBitmapBits 4-34
GpiSetBitmapDimension 4-34
GpiSetBitmapId 4-34
GpiSetCharAngle 4-35
GpiSetCharBox 4-35
GpiSetCharDirection 4-35
GpiSetCharMode 4-35
GpiSetCharSet 4-35
GpiSetCharShear 4-35
GpiSetClipPath 4-35
GpiSetClipRegion 4-36
GpiSetColor 4-36
GpiSetCp 4-36
GpiSetCurrentPosition 4-36
GpiSetDefArcParams 4-36
GpiSetDefAttrs 4-36
GpiSetDefaultViewMatrix 4-37
GpiSetDefTag 4-37
GpiSetDefViewingLimits 4-37
GpiSetDrawControl 4-37
GpiSetDrawingMode 4-37
GpiSetEditMode 4-37
GpiSetElementPointer 4-38
GpiSetElementPointerAtLabel 4-38
GpiSetGraphicsField 4-38
GpiSetInitialSegmentAttrs 4-38
GpiSetLineEnd 4-38
GpiSetLineJoin 4-38
GpiSetLineType 4-38
GpiSetLineWidth 4-39
GpiSetLineWidthGeom 4-39
GpiSetMarker 4-39
GpiSetMarkerBox 4-39
GpiSetMarkerSet 4-39
GpiSetMetaFileBits 4-39
GpiSetMix 4-39
GpiSetModelTransformMatrix 4-40
GpiSetPageViewport 4-40
GpiSetPattern 4-40
GpiSetPatternRefPoint 4-40

GpiSetPatternSet 4-40
GpiSetPel 4-40
GpiSetPickAperturePosition 4-41
GpiSetPickApertureSize 4-41
GpiSetPS 4-41
GpiSetRegion 4-41
GpiSetSegmentAttrs 4-41
GpiSetSegmentPriority 4-42
GpiSetSegmentTransformMatrix 4-42
GpiSetStopDraw 4-42
GpiSetTag 4-42
GpiSetViewingLimits 4-42
GpiSetViewingTransformMatrix 4-43
GpiStrokePath 4-43
GpiTranslate 4-43
GpiUnloadFonts 4-43
GpiUnrealizeColorTable 4-43
GpiWCBitBlt 4-44
GRADIENT 2-11
GRADIENTL 2-11
GSAM 4-33
GSAREA 4-1
GSASS 4-1
GSBBIT 4-34
GSBCOL 4-34
GSBDIM 4-34
GSBMID 4-34
GSBMIX 4-34
GSCBM 4-6
GSCD 4-35
GSCHAP 4-2
GSCLCT 4-6
GSCM 4-35
GSCOL 4-36
GSCOMM 4-4
GSCPG 4-36
GSCREG 4-4
GSCRLF 4-7
GSCS 4-35
GSDBM 4-7
GSDC 4-37
GSDCHN 4-9
GSDDYN 4-9
GSDELB 4-8
GSDELS 4-8
GSDF 4-9
GSDLE 4-7
GSDLER 4-8
GSDM 4-37
GSDREG 4-9
GSDSEG 4-9
GSDTAG 4-37
GSEDM 4-37
GSELEM 4-1
GSENDA 4-9
GSENDE 4-9
GSENDP 4-9
GSEP 4-38
GSEPLB 4-38
GSEREG 4-10
GSERSE 4-10
GSESD 4-10
GSFCLS 4-3
GSFILP 4-10
GSFLO 4-12
GSFUL 4-43
GSGET 4-11
GSIMG 4-11
GSINLB 4-11
GSLBM 4-12
GSLE 4-38
GSLJ 4-38
GSLOAD 4-14
GSLT 4-38
GSMCPY 4-4
GSMDEL 4-8
GSMFB 4-39
GSMGB 4-25
GSMIX 4-39
GSMLOD 4-12
GSMODP 4-12
GSMS 4-39
GSMSAV 4-33
GSMSS 4-39
GSOEP 4-13
GSOUTP 4-13
GSPAT 4-40
GSPATH 4-1
GSPATS 4-40
GSPOP 4-15
GSPREG 4-13
GSPUT 4-16
GSQAM 4-16
GSQATI 4-24
GSQATS 4-29
GSQBB 4-17
GSQBCO 4-16
GSQBMX 4-17
GSQBTD 4-17
GSQBTH 4-17
GSQBTP 4-17
GSQCD 4-18
GSQCLR 4-19
GSQCM 4-18
GSQCOD 4-20
GSQCOI 4-20
GSQCOL 4-20
GSQCPG 4-20
GSQCS 4-18
GSQD 4-21
GSQDBF 4-22
GSQDC 4-22
GSQDM 4-22
GSQDTA 4-21
GSQEDM 4-22
GSQEP 4-22
GSQETS 4-22
GSQF 4-23
GSQFD 4-23
GSQFM 4-23
GSQFWT 4-31
GSQKPR 4-24
GSQLCT 4-25
GSQLE 4-24
GSQLJ 4-24
GSQLT 4-24
GSQMFL 4-25
GSQMIX 4-26
GSQMS 4-25
GSQMSS 4-25
GSQNAM 4-29
GSQNC 4-26
GSQNSS 4-26
GSQPAT 4-26

GSQPRI 4-29
GSQPTS 4-27
GSQRC 4-28
GSQRGB 4-29
GSQSDW 4-30
GSQSS 4-30
GSQTAG 4-30
GSRBD 4-32
GSRCT 4-31
GSRDYN 4-32
GSRPS 4-32
GSRSPS 4-32
GSRSS 4-8
GSSATI 4-38
GSSATS 4-41
GSSBM 4-34
GSSCLP 4-35
GSSCLR 4-36
GSSCLS 4-3
GSSDEL 4-8
GSSDW 4-42
GSSEG 4-13
GSSPRI 4-42
GSSPS 4-33
GSSTRP 4-43
GSTAG 4-42
GSUCT 4-43

H

HAB 2-11
HACCEL 2-11
HANDLE 2-11
HAPP 2-11
HATOMTBL 2-11
HBITMAP 2-11
HCINFO 2-11
HDC 2-12
HENUM 2-12
HINI 2-12
HLIB 2-12
HMF 2-12
HMODULE 2-12
HMQ 2-12
HPOINTER 2-12
HPROC 2-12
HPROGARRAY 2-12
HPROGRAM 2-12
HPS 2-12
HRGN 2-12
HSEM 2-12
HSPL 2-12
HSTRUCT 2-12
HSWITCH 2-12
HVPS 2-12
HWND 2-12

I

IDENTITY 2-13
IDENTITY4 2-13
IMAGEBUNDLE 2-13
INDEX2 2-13
IPT 2-13

K

kerning 2-13
KERNINGPAIRS 2-13

L

LENGTH1 2-13
LENGTH2 2-13
LENGTH4 2-13
LINEBUNDLE 2-14
LONG 2-14

M

MARGSTRUCT 2-14
MARKERBUNDLE 2-15
MATRIX 2-15
MATRIXLF 2-15
MENUITEM 2-16
MLECTLDATA 2-16
MLE_SEARCHDATA 2-16
MPARAM 2-17
MQINFO 2-17
MRESULT 2-17

O

OFFSET2B 2-17
OVERFLOW 2-17
OWNERITEM 2-18

P

PARAM 2-18
PIBSTRUCT 2-18
PicIchg 5-1
PICONV 5-1
PicPrint 5-1
PID 2-18
PIPRT 5-1
PIX 2-18
POINT 2-18
POINTERINFO 2-19
POINTL 2-18
POINTS 2-18
PRADPR 6-1
PRCGRP 6-1
PRCPR 6-1
PRCPRG 6-1
PRDGRP 6-1
PRESDATA 2-18
PRESPARAMS 2-18
PrfAddProgram 6-1
PrfChangeProgram 6-1
PrfCloseProfile 6-1
PrfCreateGroup 6-1
PrfDestroyGroup 6-1
PrfOpenProfile 6-2
PRFPROFILE 2-18
PrfQueryDefinition 6-2
PrfQueryProfile 6-2
PrfQueryProfileData 6-2
PrfQueryProfileInt 6-3
PrfQueryProfileSize 6-3
PrfQueryProfileString 6-3

PrfQueryProgramCategory 6-4
PrfQueryProgramHandle 6-4
PrfQueryProgramTitles 6-4
PrfRemoveProgram 6-4
PrfReset 6-5
PrfWriteProfileData 6-5
PrfWriteProfileString 6-5
PROC 2-18
PROGCATEGORY 2-18
PROGDETAILS 2-18
PROGRAMENTRY 2-19
PROGTITLE 2-19
PROGTYPE 2-19
PROPERTY2 2-19
PROPERTY4 2-19
PROPPR 6-2
PRQDEF 6-2
PRQPC 6-4
PRQPDA 6-2
PRQPHD 6-4
PRQPIN 6-3
PRQPR 6-2
PRQPST 6-3
PRQPSZ 6-3
PRQPTI 6-4
PRRES 6-5
PRRPRG 6-4
PRWPDA 6-5
PRWPST 6-5

Q

QMOPENDATA 2-19
QMSG 2-20

R

RECT 2-20
RECTL 2-20
RESID 2-20
RGB 2-20
RGNRECT 2-20
ROF 2-21
ROL 2-21

S

SBCDATA 2-21
SEGOFF 2-21
SFACTORS 2-21
SHORT 2-21
SIZEF 2-21
SIZEL 2-21
SIZEROF 2-21
SIZEROL 2-21
SMABT 7-1
SMCLS 7-1
SMEDOC 7-1
SMHSTRUCT 2-21
SMOPEN 7-1
SMSDOC 7-1
SMWRT 7-2
SPINST 7-2
SplQmAbort 7-1
SplQmClose 7-1
SplQmEndDoc 7-1
SplQmOpen 7-1
SplQmStartDoc 7-1
SplQmWrite 7-2
SplQpInstall 7-2
SplQpQueryDt 7-2
SPQDT 7-2
STORAGE 2-22
STR 2-22
STRCOND 2-22
STRL 2-22
STRLLIST 2-22
structure utilities 1-4
STR16 2-22
STR32 2-22
STR64 2-22
STR8 2-22
SWBLOCK 2-22
SWCNTRL 2-23
SWENTRY 2-23
SWP 2-23

T

TID 2-23
TIME 2-23
TRACKINFO 2-24

U

UCHAR 2-24
ULONG 2-24
USERBUTTON 2-24
USHORT 2-24
utilities
 bit 1-4
 composed-integer 1-4
 structure 1-4
utility functions 1-4

W

WCCRD 8-6
WCDDP 8-9
WCDELD 8-11
WCEDLI 8-15
WCGETD 8-17
WCISMG 8-20
WCLD 8-23
WCLM 8-23
WCLMD 8-12
WCMBOX 8-25
WCMDP 8-25
WCPD 8-29
WCPQM 8-28
WCQDIS 8-33
WCQMPO 8-34
WCQMTI 8-34
WCQSTA 8-37
WCSDI 8-48
WCSDIM 8-45
WDCLS 3-1
WDEPI 8-14
WDESC 3-1
WDOPEN 3-1
WDPDM 3-2
WDQCAP 3-2

WDQDN 3-2
WDQHC 3-3
WIBEWN 8-1
WIBORD 8-13
WIBPNT 8-2
WICCBD 8-3
WICHLD 8-21
WICPIN 8-7
WICPOI 8-7
WICR 8-4
WICRCT 8-2
WICRT 8-9
WICRTS 8-8
WIDBMP 8-12
WIDEL 8-11
WIDTH4 2-25
WIDTXT 8-13
WIECB 8-14
WIECF 8-15
WIEEW 8-14
WIEPNT 8-14
WIEREC 8-15
WIEUPR 8-15
WIEWU 8-14
WIFLSH 8-16
WIFRCO 8-6
WIFRCT 8-16
WIGCTI 8-17
WIGMNP 8-18
WIGWIN 8-18
WILKVR 8-24
WILKWN 8-24
WIMAPW 8-25
WIMWID 8-26
WinAddAtom 8-1
WinAddProgram 8-1
WinAddSwitchEntry 8-1
WinAlarm 8-1
WinBeginEnumWindows 8-1
WinBeginPaint 8-2
WinBroadcastMsg 8-2
WinCalcFrameRect 8-2
WinCallMsgFilter 8-2
WinCancelShutdown 8-3
WinChangeSwitchEntry 8-3
WinCloseClipbrd 8-3
WinCompareStrings 8-3
WinCopyAccelTable 8-3
WinCopyRect 8-4
WinCpTranslateChar 8-4
WinCpTranslateString 8-4
WinCreateAccelTable 8-5
WinCreateAtomTable 8-5
WinCreateCursor 8-5
WinCreateDataStructure 8-6
WinCreateDlg 8-6
WinCreateFrameControls 8-6
WinCreateGroup 8-7
WinCreateMsgQueue 8-7
WinCreatePointer 8-7
WinCreatePointerIndirect 8-7
WinCreateStdWindow 8-8
WinCreateSwitchEntry 8-8
WinCreateWindow 8-9
WinDefDlgProc 8-9
WinDefWindowProc 8-9
WinDeleteAtom 8-10
WinDeleteLibrary 8-10
WinDeleteProcedure 8-10
WinDestroyAccelTable 8-10
WinDestroyAtomTable 8-10
WinDestroyCursor 8-10
WinDestroyDataStructure 8-11
WinDestroyMsgQueue 8-11
WinDestroyPointer 8-11
WinDestroyWindow 8-11
WinDismissDlg 8-11
WinDispatchMsg 8-11
WinDlgBox 8-12
WinDrawBitmap 8-12
WinDrawBorder 8-13
WinDrawPointer 8-13
WinDrawText 8-13
WinEmptyClipbrd 8-14
WinEnablePhysInput 8-14
WinEnableWindow 8-14
WinEnableWindowUpdate 8-14
WinEndEnumWindows 8-14
WinEndPaint 8-14
WinEnumClipbrdFmts 8-15
WinEnumDlgItem 8-15
WinEqualRect 8-15
WinExcludeUpdateRegion 8-15
WinFillRect 8-16
WinFindAtom 8-16
WinFlashWindow 8-16
WINFLR 8-19
WinFocusChange 8-16
WinFreeErrorInfo 8-16
WinFreeMsg 8-17
WinGetClipPS 8-17
WinGetCurrentTime 8-17
WinGetDlgMsg 8-17
WinGetErrorInfo 8-17
WinGetKeyState 8-18
WinGetLastError 8-18
WinGetMinPosition 8-18
WinGetMsg 8-18
WinGetNextWindow 8-18
WinGetPhysKeyState 8-19
WinGetPS 8-19
WinGetScreenPS 8-19
WinGetSysBitmap 8-19
WinInflateRect 8-19
WinInitialize 8-19
WinInSendMsg 8-20
WinInstStartApp 8-20
WinIntersectRect 8-20
WinInvalidateRect 8-21
WinInvalidateRegion 8-21
WinInvertRect 8-21
WinIsChild 8-21
WinIsRectEmpty 8-22
WinIsThreadActive 8-22
WinIsWindow 8-22
WinIsWindowEnabled 8-22
WinIsWindowShowing 8-22
WinIsWindowVisible 8-22
WinLoadAccelTable 8-23
WinLoadDlg 8-23
WinLoadLibrary 8-23
WinLoadMenu 8-23
WinLoadPointer 8-24
WinLoadProcedure 8-24

WinLoadString 8-24
WinLockVisRegions 8-24
WinLockWindow 8-24
WinLockWindowUpdate 8-25
WinMapDlgPoints 8-25
WinMapWindowPoints 8-25
WinMessageBox 8-25
WinModifyDataStructure 8-26
WinMsgMuxSemWait 8-26
WinMsgSemWait 8-26
WinMultWindowFromIDs 8-26
WinNextChar 8-27
WinOffsetRect 8-27
WinOpenClipbrd 8-27
WinOpenWindowDC 8-27
WinPeekMsg 8-28
WinPostMsg 8-28
WinPostQueueMsg 8-28
WinPrevChar 8-29
WinProcessDlg 8-29
WinPtInRect 8-29
WinQueryAccelTable 8-29
WinQueryActiveWindow 8-29
WinQueryAnchorBlock 8-30
WinQueryAtomLength 8-30
WinQueryAtomName 8-30
WinQueryAtomUsage 8-30
WinQueryBits 8-30
WinQueryBitsUnderMask 8-31
WinQueryCapture 8-31
WinQueryClassInfo 8-31
WinQueryClassName 8-31
WinQueryClipbrdData 8-31
WinQueryClipbrdFmtInfo 8-32
WinQueryClipbrdOwner 8-32
WinQueryClipbrdViewer 8-32
WinQueryCp 8-32
WinQueryCpList 8-32
WinQueryCursorInfo 8-32
WinQueryDataStructure 8-33
WinQueryDefinition 8-33
WinQueryDesktopWindow 8-33
WinQueryDlgItemShort 8-33
WinQueryDlgItemText 8-34
WinQueryDlgItemTextLength 8-34
WinQueryFocus 8-34
WinQueryMsgPos 8-34
WinQueryMsgTime 8-34
WinQueryObjectWindow 8-34
WinQueryPointer 8-35
WinQueryPointerInfo 8-35
WinQueryPointerPos 8-35
WinQueryPresParam 8-35
WinQueryProfileData 8-36
WinQueryProfileInt 8-36
WinQueryProfileSize 8-36
WinQueryProfileString 8-37
WinQueryProgramTitles 8-37
WinQueryQueueInfo 8-37
WinQueryQueueStatus 8-37
WinQuerySessionTitle 8-38
WinQuerySwitchEntry 8-38
WinQuerySwitchHandle 8-38
WinQuerySwitchList 8-38
WinQuerySysColor 8-38
WinQuerySysModalWindow 8-39
WinQuerySysPointer 8-39
WinQuerySystemAtomTable 8-39
WinQuerySysValue 8-39
WinQueryTaskSizePos 8-39
WinQueryTaskTitle 8-40
WinQueryUpdateRect 8-40
WinQueryUpdateRegion 8-40
WinQueryValue 8-40
WinQueryVersion 8-40
WinQueryWindow 8-41
WinQueryWindowDC 8-41
WinQueryWindowLockCount 8-41
WinQueryWindowPos 8-41
WinQueryWindowProcess 8-41
WinQueryWindowPtr 8-41
WinQueryWindowRect 8-42
WinQueryWindowText 8-42
WinQueryWindowTextLength 8-42
WinQueryWindowULong 8-42
WinQueryWindowUShort 8-42
WinRegisterClass 8-43
WinRegisterUserDatatype 8-43
WinRegisterUserMsg 8-43
WinRegisterWindowDestroy 8-43
WinReleaseHook 8-44
WinReleasePS 8-44
WinRemovePresParam 8-44
WinRemoveSwitchEntry 8-44
WinScrollWindow 8-45
WinSendDlgItemMsg 8-45
WinSendMsg 8-45
WinSetAccelTable 8-46
WinSetActiveWindow 8-46
WinSetBits 8-46
WinSetBitsUnderMask 8-46
WinSetCapture 8-46
WinSetClassMsgInterest 8-47
WinSetClipbrdData 8-47
WinSetClipbrdOwner 8-47
WinSetClipbrdViewer 8-47
WinSetCp 8-47
WinSetDlgItemShort 8-48
WinSetDlgItemText 8-48
WinSetFocus 8-48
WinSetHook 8-48
WinSetKeyboardStateTable 8-49
WinSetMsgInterest 8-49
WinSetMsgMode 8-49
WinSetMultWindowPos 8-49
WinSetOwner 8-49
WinSetParent 8-50
WinSetPointer 8-50
WinSetPointerPos 8-50
WinSetPresParam 8-50
WinSetRect 8-51
WinSetRectEmpty 8-51
WinSetSynchroMode 8-51
WinSetSysColors 8-51
WinSetSysModalWindow 8-52
WinSetSysValue 8-52
WinSetValue 8-52
WinSetWindowBits 8-52
WinSetWindowPos 8-53
WinSetWindowPtr 8-53
WinSetWindowText 8-53
WinSetWindowULong 8-53
WinSetWindowUShort 8-54
WinShowCursor 8-54

WinShowPointer 8-54
WinShowTrackRect 8-54
WinShowWindow 8-54
WinStartTimer 8-55
WinStopTimer 8-55
WinSubstituteStrings 8-55
WinSubtractRect 8-55
WinSwitchToProgram 8-56
WinTerminate 8-56
WinTerminateApp 8-56
WinTrackRect 8-56
WinTranslateAccel 8-56
WINTSR 8-20
WinUnionRect 8-57
WinUpdateWindow 8-57
WinUpper 8-57
WinUpperChar 8-57
WinValidateRect 8-58
WinValidateRegion 8-58
WINVER 8-21
WINVR 8-21
WINVRG 8-21
WinWaitMsg 8-58
WinWindowFromDC 8-58
WinWindowFromID 8-58
WinWindowFromPoint 8-59
WinWriteProfileData 8-59
WinWriteProfileString 8-59
WIOFFR 8-27
WIOPCB 8-27
WIOPDC 8-27
WIPTRC 8-29
WIQCBD 8-31
WIQCBF 8-32
WIQCBO 8-32
WIQCBV 8-32
WIQDTW 8-33
WIQENA 8-22
WIQHAB 8-30
WIQHDC 8-58
WIQHID 8-58
WIQOBJ 8-34
WIQPOS 8-41
WIQPRO 8-41
WIQRCT 8-42
WIQSBM 8-19
WIQSCO 8-38
WIQSHO 8-22
WIQSMW 8-39
WIQTXL 8-42
WIQTXT 8-42
WIQURC 8-40
WIQURG 8-40
WIQVIS 8-22
WIQWDC 8-41
WIQWIN 8-41
WIQWLC 8-41
WIQWPT 8-41
WIQWUL 8-42
WIQWUS 8-42
WIREMP 8-22
WISCBD 8-47
WISCBO 8-47
WISCBV 8-47
WISCOL 8-51
WISCRL 8-45
WISCUR 8-54
WISENA 8-14
WISHOW 8-54
WISMPS 8-49
WISOWN 8-49
WISPAR 8-50
WISPOS 8-53
WISRE 8-51
WISREC 8-51
WISSMW 8-52
WISTAT 8-55
WISTOT 8-55
WISTR 8-54
WISTXT 8-53
WISUBR 8-55
WISWBT 8-52
WISWIN 8-22
WISWPT 8-53
WISWUL 8-53
WISWUS 8-54
WITREC 8-56
WIUNRC 8-57
WIUPDT 8-57
WIVREC 8-58
WIVRGN 8-58
WIWFP 8-59
WMADPR 8-1
WMALRM 8-1
WMASWE 8-1
WMBRMG 8-2
WMCACT 8-5
WMCAT 8-3
WMCATT 8-5
WMCCUR 8-5
WMCGRP 8-7
WMCMFI 8-2
WMCMQ 8-7
WMCRSE 8-8
WMCSD 8-3
WMCSWE 8-3
WMDACT 8-10
WMDATT 8-10
WMDCUR 8-10
WMDELA 8-10
WMDISP 8-11
WMDLIB 8-10
WMDMQ 8-11
WMDPOI 8-11
WMDPR 8-10
WMDRPT 8-13
WMDWP 8-9
WMFERI 8-16
WMFNAT 8-16
WMFOCC 8-16
WMGECP 8-32
WMGEI 8-17
WMGET 8-18
WMGLE 8-18
WMGPKS 8-19
WMINIT 8-19
WMISAP 8-20
WMKEYS 8-18
WMLACT 8-23
WMLKWU 8-25
WMLLIB 8-23
WMLPOI 8-24
WMLPR 8-24
WMLSTR 8-24

WMMINT 8-49
WMMSWT 8-26
WMPEEK 8-28
WMPOST 8-28
WMQACC 8-29
WMQACT 8-29
WMQALN 8-30
WMQANM 8-30
WMQAUS 8-30
WMQCAP 8-31
WMQCLI 8-31
WMQCLN 8-31
WMQCPL 8-32
WMQCUR 8-32
WMQDEF 8-33
WMQDIL 8-34
WMQDIT 8-34
WMQFOC 8-34
WMQPDA 8-36
WMQPIN 8-36
WMQPP 8-35
WMQPST 8-37
WMQPSZ 8-36
WMQPTD 8-35
WMQPTI 8-37
WMQPTP 8-35
WMQPTR 8-35
WMQQI 8-37
WMQSAT 8-39
WMQSEN 8-38
WMQSHD 8-38
WMQSLI 8-38
WMQSPT 8-39
WMQSTI 8-38
WMQSYS 8-39
WMQTSP 8-39
WMQTTI 8-40
WMQVER 8-40
WMRCL 8-43
WMREWD 8-43
WMRHK 8-44
WMRPP 8-44
WMRSWE 8-44
WMRUD 8-43
WMRUM 8-43
WMSACC 8-46
WMSACT 8-46
WMSCAP 8-46
WMSCMI 8-47
WMSDIT 8-48
WMSECP 8-47
WMSEND 8-45
WMSFOC 8-48
WMSHOO 8-48
WMSHPT 8-54
WMSKST 8-49
WMSMM 8-49
WMSPP 8-50
WMSPTP 8-50
WMSPTR 8-50
WMSSM 8-51
WMSSYS 8-52
WMSTOP 8-56
WMSTP 8-56
WMSWT 8-26
WMTAC 8-56
WMTERM 8-56
WMTHRA 8-22
WMWAIT 8-58
WMWPDA 8-59
WMWPST 8-59
WNDPARAMS 2-25
WNDPROC 2-25
WPGCPS 8-17
WPGPS 8-19
WPGSPS 8-19
WPOINT 2-25
WPREPS 8-44
WUADAT 8-1
WUCDS 8-6
WUCOMP 8-3
WUDDS 8-11
WUFMSG 8-17
WUMDS 8-26
WUNXTC 8-27
WUPCH 8-57
WUPPER 8-57
WUPREC 8-29
WUQBIT 8-30
WUQBM 8-31
WUQDS 8-33
WUQVAL 8-40
WUSBIT 8-46
WUSBM 8-46
WUSUBS 8-55
WUSVAL 8-52
WUTRCH 8-4
WUTRST 8-4

X

XYWINSIZE 2-25

Special Characters

→ 1-3

IBM United Kingdom
International Products Limited
PO Box 41, North Harbour
Portsmouth, PO6 3AU
England

Printed in Denmark by Rosendahl - Esbjerg

